昭和59年2月28日国鉄首都特別扱承認雑誌第7403号　昭和59年4月1日発行(毎月1日発行)第2巻第2号

PERSONAL COMPUTER MAGAZINE
月刊オー！エフエム

Oh!FM

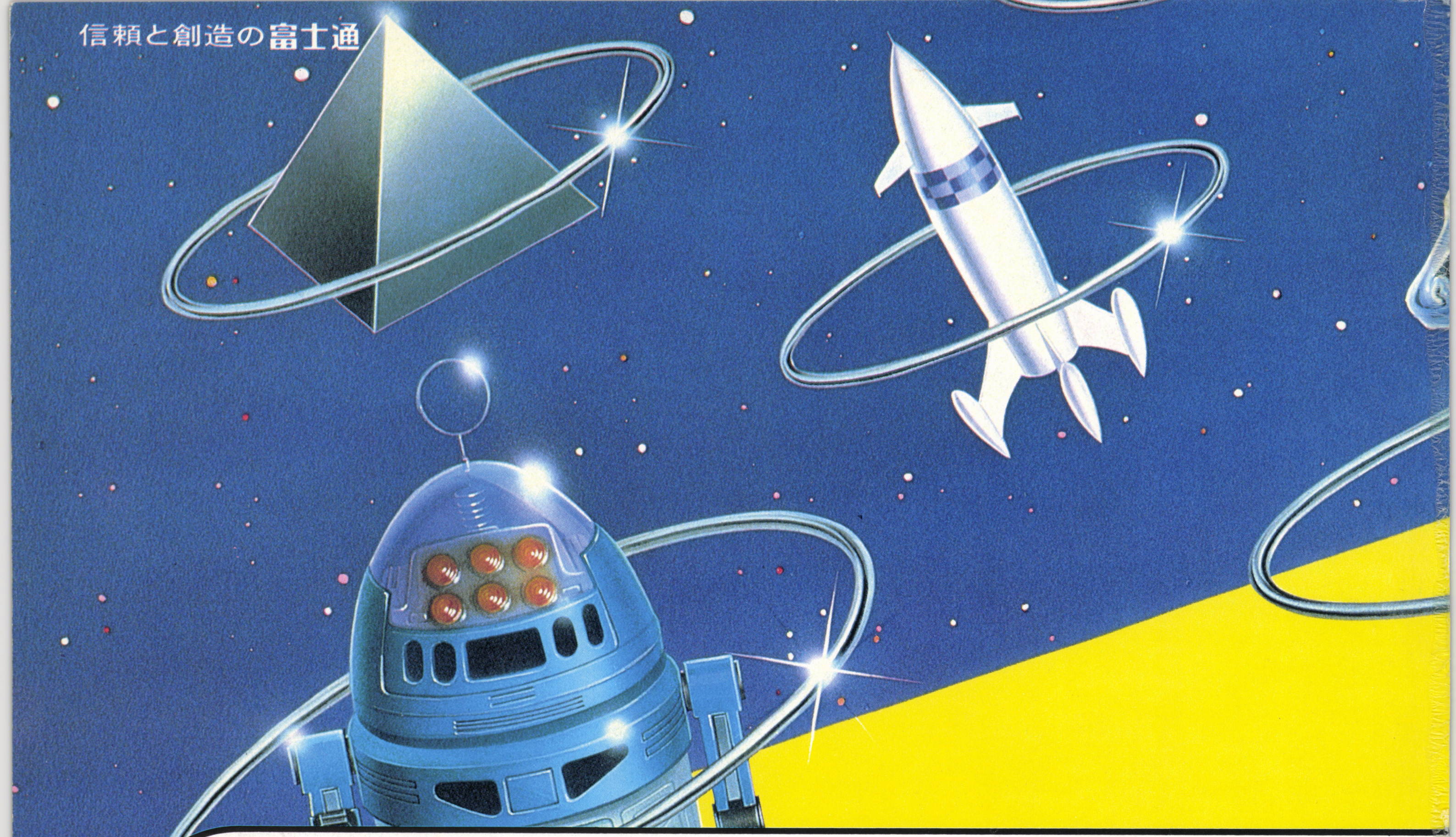
FM-7 8 11

特集　虫取り専科

FM-7を解析する
アドベンチャゲーム開発ツール
6809機械語マスタ'84
FM-11のニューフェース
FM-11BS,AD2

1984
4

信頼と創造の富士通



FM SPIRIT. 満開、

夜空に輝く宇宙船、ナスカの地上絵、ソロモンの秘宝。
いくつになっても、どんな時でも
興奮させてくれるモノなら大歓迎。
そんな敏感なココロがFMスピリッツ。

友だちづきあいできるパソコン

今、話題のMSX規格に対応。操作はカンタン、ROMカートリッジをスロットへ入れるだけ。手にしたその日からゲームをはじめ、学習、家事にと幅広く友だちづきあいのできるパソコンです。

●RFコンバータを内蔵、家庭用テレビの画面がそのままディスプレイに早がわり。

●多彩な16色のカラーグラフィック表示。

●8オクターブ3重和音のサウンド機能。

●FM-7と接続することにより、MSXだけでは味わえないスピード感あふれるグラフィック、そしてダブルPSGによるマルチサウンドが実現。

MSX MSXマークはマイクロソフトの商標です。

FM-X エックス

¥49,800（本体価格）

新発売

富士通株式会社●半導体統轄営業部(03)502-0161●札幌営業所(011)271-4311●東北営業所(0222)64-2131●金沢営業所(0762)63-7621●長野営業所(0262)26-8222●静岡営業所(0542)54-9131●名古屋営業所(052)201-8611●大阪営業所(06)344-1101●広島営業所(082)221-2288●九州営業所(092)411-6311

ボクらの興奮ゴコロ

豊富なソフトで、ますます親しみやすくなるパソコンFMシリーズ。

テクノロジーで育つボクらの感覚は、こんなマシンを見逃せない。

RUN THE FM!

このスピリッツを体験しないか。

ハードで興奮、ソフトで感激

発売以来ベストセラー。充実のグラフィック機能、多彩な周辺装置など、ハードは万全。加えて、教育用からホビーまで、あらゆるジャンルで、つぎつぎと登場するエキサイティングソフト。全国のパソコンファン興奮の一台です。

●CPU、68B09を2基搭載、群を抜く処理速度、スピーディなグラフィック。

●教育用言語として注目されるFM Logo（FM-7用）を新発売。（カセット版￥13,000、ミニフロッピィ版￥16,000）

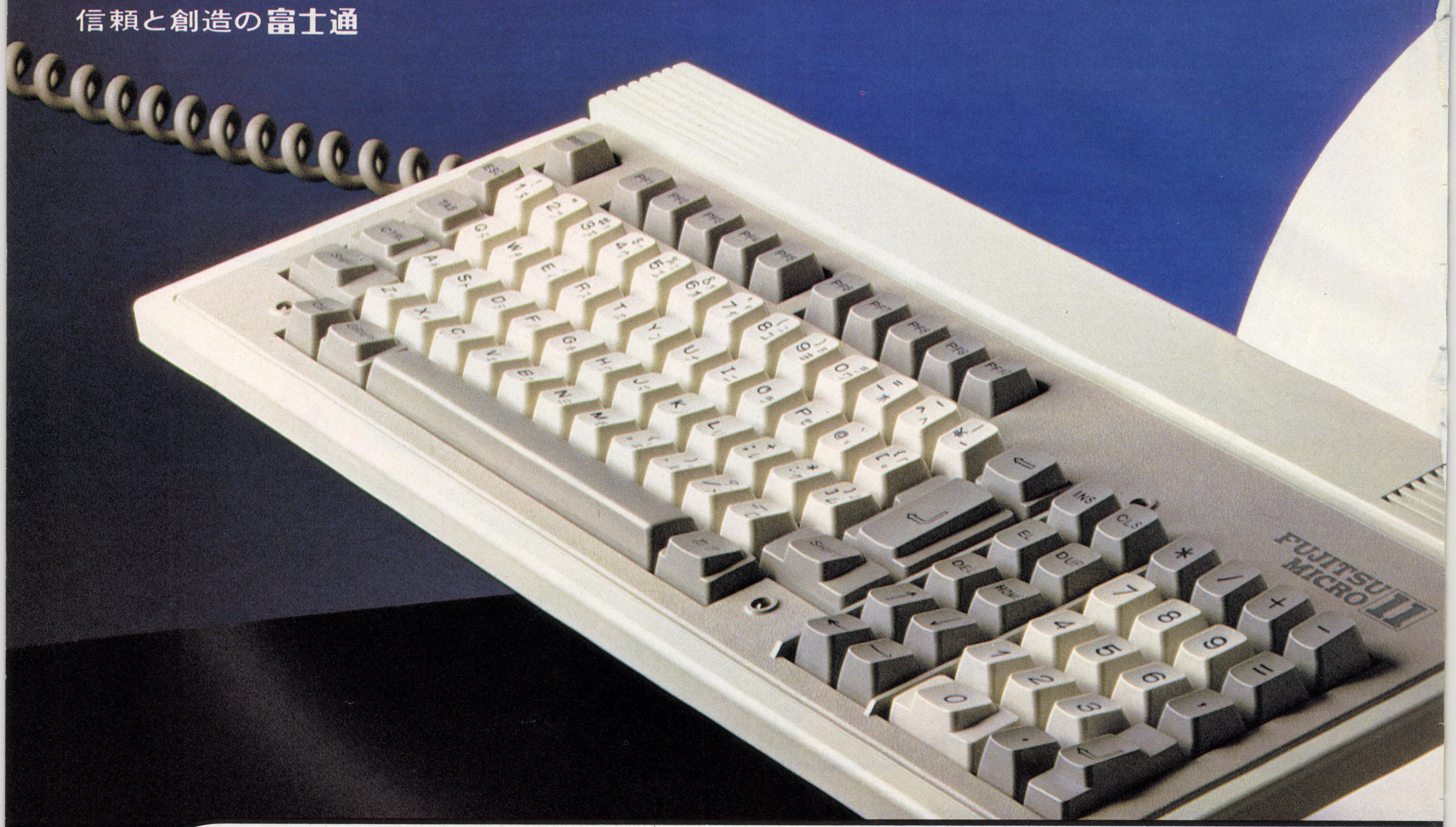
●熱転写プリンタ、I/O拡張ボックス等、教育・ビジネス・マニア向けの周辺機器が続々登場。

FM-7 セブン　**¥126,000**（本体価格 簡易言語ソフト付）

高級ホビーからビジネスまでの多才パソコン。**FM-8** エイト ￥218,000（本体価格）

富士通株式会社・半導体統轄営業部
〒105 東京都港区虎ノ門2-3-13 ☎(03)502-0161

[広]1



FM SPIRIT. 実感、超え

最新のOAを、まるでペンを走らせるように気楽に使いこなしてしまう

そんなビジネスマンが増えています。

先端技術を有能なパートナーに

そんな粋なFMスピリッツが、今、注目を集めます。

OS-9を採用、ビジネスもマルチ感覚

究極の8ビットCPU“68B09”を、さらにパワーアップするオペレーティングシステムOS-9™を採用。複数の処理を同時に実行。さらに一つの画面を分割し、それぞれに独立した処理の表示が可能です。驚異のマルチ機能で、オフィスの能率は飛躍的にアップ。

●ミニコンのOSの主流、UNIX™の特長を受け継ぐ、マルチ・タスク、マルチ・ウィンドウ等の機能を持つ、OS-9を標準添付。

●大容量1MBの5インチフロッピィディスクドライブを1基実装。

●漢字ROM(JIS第一水準)を標準実装。

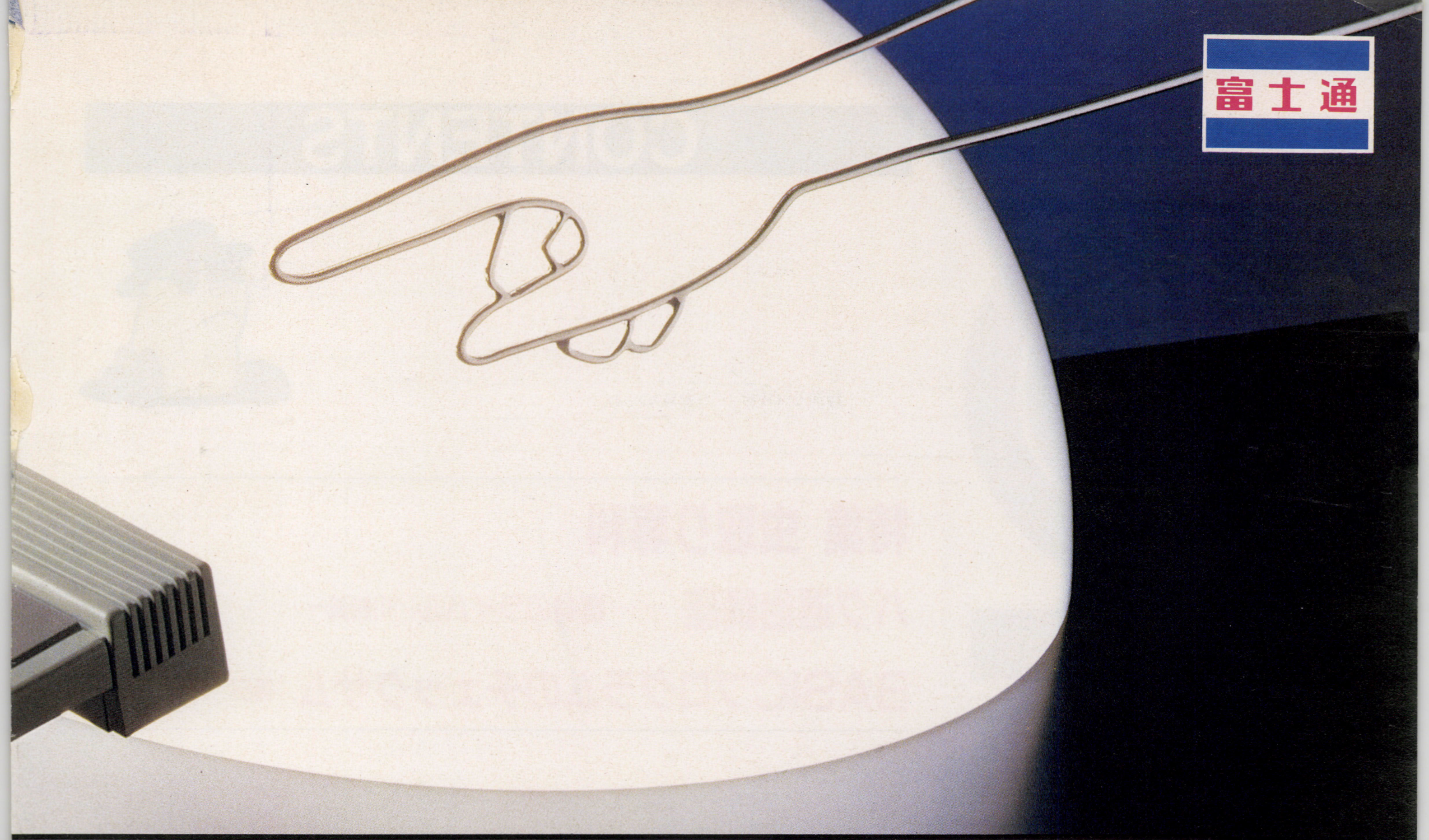
●高性能、日本語ワードプロセッサFM-JWP/11を添付。

FM-11 イレブン **AD2** ¥298,000
(本体価格・日本語ワープロソフト付)

新登場

※OS-9はマイクロウェア社、UNIXはベル研究所の登録商標です。

マイコンスカイラブ：FMシリーズのハードからソフトまで一挙に展示実演。あなたのパソコンのコンサルタントとしてご活用ください。●東京・虎ノ門(03)591-1091／591-2561●東京・秋葉原(03)251-1448／251-1449●札幌〈時計台ビル〉(011)222-5476〈丸井今井〉(011)241-4185●仙台(0222)66-8711●名古屋(052)221-6016●大阪(06)344-7628／341-0486●広島(082)247-3949●福岡〈開設準備室〉(092)471-7203



てる人のビジネスごころ

グレードアップして新登場のパソコンFM-11。

先端のOSを搭載し、16ビット、8ビットの2機種で

幅広く、オフィスの期待にこたえます。

群を抜くパフォーマンスで、今、FMスピリッツあふれるビジネスマンを強力にサポート。

●写真はFM-11 BSです。(CRTは新タイプMB 27315)

1MBの大容量5インチFDを2基搭載

ビジネスユースとしての、16ビットマシンを、ハードウェア及びソフトウェアの強化により、さらに本格的な実務に対応させました。大容量の記憶装置、通常業務には欠かせない日本語対応ソフトの充実など、すぐれたコストパフォーマンスを実現しました。

●大容量1MBの5インチフロッピィディスクドライブを2基実装。しかもRAMエリアは256KBを標準実装。

●辞書サポートにより、日本語処理機能を強化したCP/M-86®、F-BASIC86を添付。

●漢字ROM(JIS才一水準)を標準実装。

●CP/M-86®配下の高性能、日本語ワードプロセッサJWORDを添付。

FM-11 イレブン **BS** **¥398,000**
(本体価格・日本語ワープロソフト付)

※CP/M-86はディジタルリサーチ社の登録商標です。

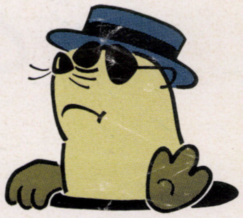
富士通株式会社●半導体統轄営業部(03)502-0161●札幌営業所(011)271-4311●東北営業所(0222)64-2131●金沢営業所(0762)63-7621●長野営業所(0262)26-8222●静岡営業所(0542)54-9131●名古屋営業所(052)201-8611●大阪営業所(06)344-1101●広島営業所(082)221-2288●九州営業所(092)411-6311

PERSONAL COMPUTER MAGAZINE

Oh!FM

1984・4

CONTENTS



特集 虫取り専科

バグ退治講座——宿命のライバル／川村誠一 …… 25

BASICプログラムのチェックサム／園田 茂 …… 41

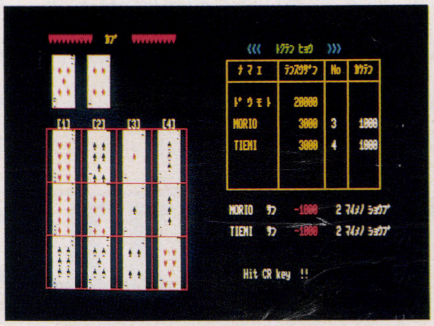
使い勝手のよいマシン語モニタ／高橋良之 …… 73

FM-7を解析する
F-BASIC V3.0のデータ構造とアルゴリズム／Q-Laboratory …… 128

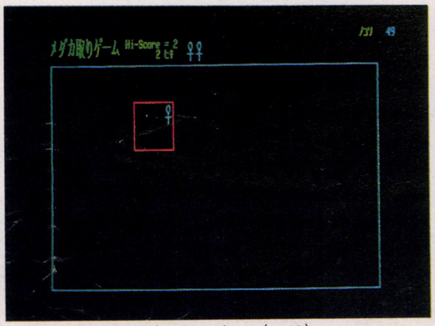
FM-11のニューフェース
FM-11BS，AD2／西村義孝 …… 18

君もアドベンチャゲームを作ってみよう
アドベンチャゲーム開発ツール／米村英明 …… 105

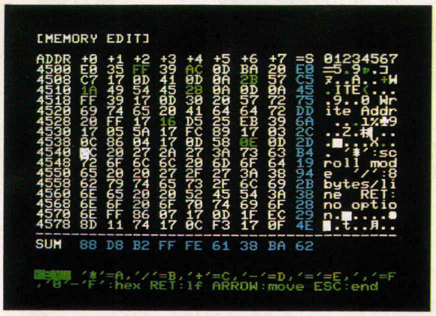
ゲームリスト パルコム「おいちょかぶ」 …… 97



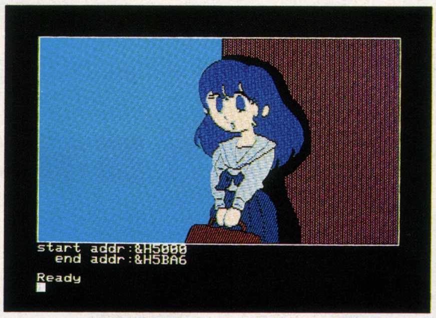
おいちょかぶ(p.97)



メダカ取りゲーム(p.43)



マシン語モニタ(p. 73)



アドベンチャゲーム開発ツール(p. 105)

新連載

ビジネス用BASICプログラムジェネレータ
【1】BASICプログラムを作るプログラムの作り方／高木征蔵……120

6809機械語マスタ'84
第1回　マシン語ってなにものだ／鶴岡哲明……91

マシン語ゲームの作り方[2]／笠作貴弥……56

続・BIOSの使い方[2]／古田　昌……114

6809機械語の書き方[後]／草津酔床……50

はじめて握るはんだごて　PART 3
オーディオタイマの製作 I／志沢雅彦……63

WELCOME TO LOGO ようこそFMロゴへ／Thabio La Baugh……138

ショートプログラム集……43

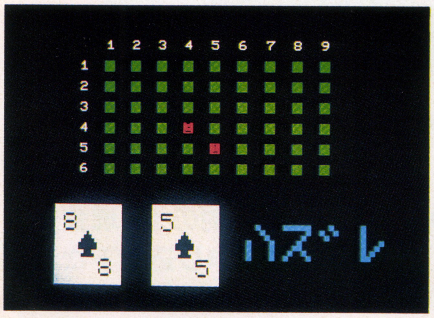
SOFT REVIEW……21

Q&A……137
READER'S AREA……47
虫つくろい……49
ほりでぇくんのパッチング……49
バックナンバー情報……104
プレゼント情報……136
広告索引……168
EDITOR'S ROOM……168

COVER STORY
Hi／Baby　　Tea Spoon　木下　淳博

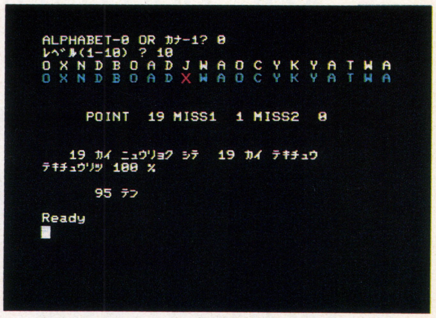
「おい，なんか食たくない？」「んー，そういやハラへったな。ちょうどガソリンも少ないし，次でひと休みするか」
「ちょっと待った／ブレーキ　ブレーキ／」
「なんだよ急に」
「ギャルの車がエンコしてるんだよ」
「ナーイス，それを早く言ってくれよ。Hi／Baby／」

・CP/M, CP/M-86, MP/MはDigital Research社
・MS-DOSはMicrosoft社
・FLEXはTSC社
・UNIXはBell Laboratories
・UCSD p-Systemはカルフォルニア大学理事会
・OS-9はMICROWARE社
・SB-80, SB-86はLIFEBOAT ASSOCIATES
・WORDSTAR, MAILMERGE, SPELLSTAR, WORDMASTER, CALCSTAR, DATASTAR, SUPERSORT, INFOSTARはMicro Pro社
の各メーカーの登録商標です。
・MSXマークはマイクロソフト社の商標です。
その他プログラム名，システム名，CPU名は一般に各開発メーカーの登録商標です。
本文中では"TM"，"Ⓡ"マークは明記していません。
本誌に掲載されたすべてのプログラムは，個人で使用する以外著作権法上，無断複製を禁じられています。



神経衰弱ゲーム(p. 44)

ポーカーゲーム(p. 45)

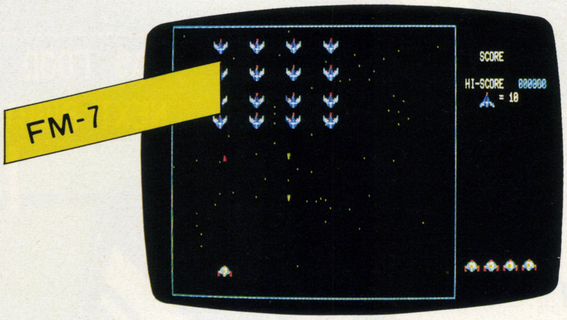


タイプ練習(p. 46)

いま話題を呼んでいる!

A.D.A. soft はByte Shopオリジナルです。

■君だけのオリジナル・ゲームが 簡単にできる!

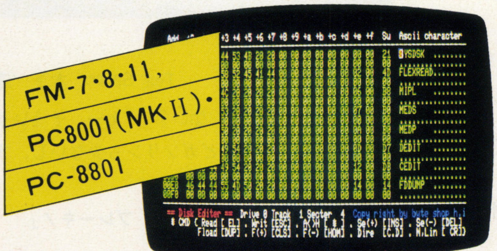


ゲームプロセッサー **GP-7**

テープ版 ¥4,200　Disk版 ¥5,200

GP-7はパターンエディターと簡易型コンパイラーを備えた「ゲームソフトを作るため」のプログラムです。BASICのテキスト上に簡単な命令を書くだけで、オールマシン語(サブCPU直接制御)のスクロールゲーム(インベーダーやギャラクシアン類)を作ることができ、完成したプログラムは独立したものとしてファイルに保存できます。(アプリケーションゲーム2本付き)

■1枚のディスケットでFM・PCシリーズが使える!



だいじょうぶマイフレンド

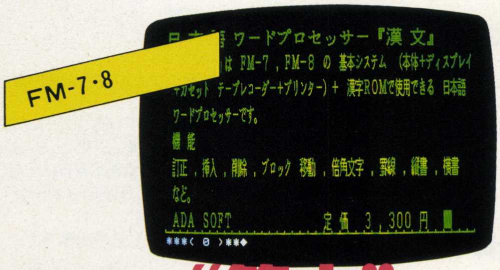
ディスクユーティリティ 定価¥6,800

FMシリーズの各プログラム
- FMシリーズのシステムDISKの作成及びVolcopy ProgramとSystem copyプログラム
- FLEX Binary Fileを読むプログラムとFLEXのディスケットのSector Dumpを行うプログラム
- FM-7の裏RAMをDiskと同じように使うプログラム
- FM-7, 8, 11用のメモリーエディター
- FM-7, 8, 11用のディスクエディター
- FM-7, 8, 11用のID表示付Volcopyプログラム
- FM-7, 8, 11用Disk Directory Dump
- 不用なプログラム及びファイルをKillするプログラム
- FM⇄FM, PC⇄PC, FM⇄PCのファイルコンバーター

PCシリーズの各プログラム
- PC-8001 & PC-8031-2W(80S31)用DISKエディター及びPC-8031内のメモリーエディターです。トラックリード, トラックライト付でPC-8031内の指定アドレスに実行を移すことができます
- メモリーエディター
- File Dump
- Program & File COPY
- プログラムとファイルの使用クラスターをDump
- DISKのFormat & BACKUP
- DiskのFATの使用状態をDump
- Fileの属性のSet, Reset, Killを行う

※PC-8801もしくはPC-8001mk2はN-BASICモードで使用できる。

■ディスクを持っている人も買う! これほんと?

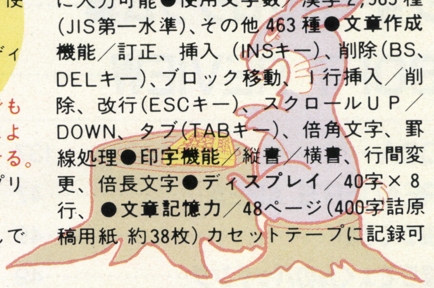


日本語ワードプロセッサ **"簡文"** テープ版

定価¥3,300

特長
- FM-7,8の基本システム(本体漢ROM付・ディスプレイ・カセットレコーダー・シリアルドットプリンター)のみで使用することができる。
- INS, DELなどのキーがBASICのエディター同様の機能を持つ。
- 漢字の呼び出しは音訓いずれからでも可能、語を品詞別に入力することによりよりスムーズに目的の漢字を捜せる。
- シリアルドットプリンターで漢字プリンターと同品質の印字が得られる。
- すべてマシン語でプログラムを組んであり高速処理ができる。

システム仕様
●入力方式/かな漢字変換(文字単位、品詞別)動詞については活用語尾とともに入力可能●使用文字数/漢字2,965種(JIS第一水準)、その他463種●文章作成機能/訂正、挿入(INSキー)、削除(BS、DELキー)、ブロック移動、1行挿入/削除、改行(ESCキー)、スクロールUP/DOWN、タブ(TABキー)、倍角文字、罫線処理●印字機能/縦書/横書、行間変更、倍長文字●ディスプレイ/40字×8行、●文章記憶力/48ページ(400字詰原稿用紙 約38枚)カセットテープに記録可



▶ ザ・ペンギンダンス	¥3,300
▶ AGL-2001F(C)	¥3,300
▶ AGL-2001F(D)	¥4,800

近日発売

フロンティア09
OS-9用スクリーンエディター
¥19,800

技術者募集

8bit,16bit,ハード, ソフトウェア技術経験者〝あなたの能力を十分に生かせる〟

TEL 03-257-6309

A.D.A.オフィス

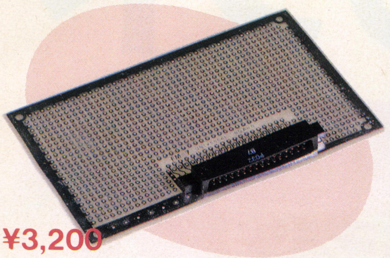
〒101 東京都千代田区神田須田町2―23―1
芝崎ビル5F(バイトショップ・サポート内)

通販御希望の方は、送料500円を同封の上、現金書留にてお申し込み下さい。

エイ・デイ・エイ

A.D.A. まかせなさい!!

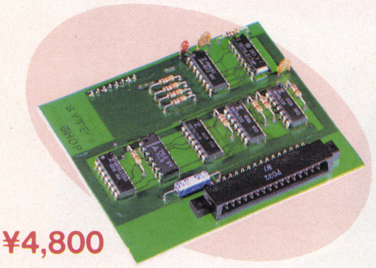
FM-7のアプリケーションは



¥3,200

汎用ユニバーサルボード ADA-7001

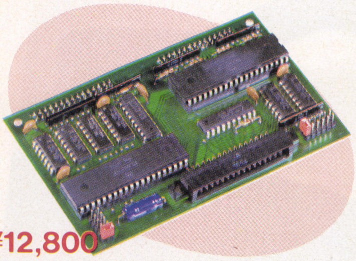
ユーザが自由な回路をFM-7 オプションスロット内に組込めます TTLIC20コ相当のユニバーサルボードです ガラスエポキシ、スルーホール、基板サイズ 125×77、32P・40P共用可能、2.54ミリピッチ



¥4,800

アナログボード ADA-7002

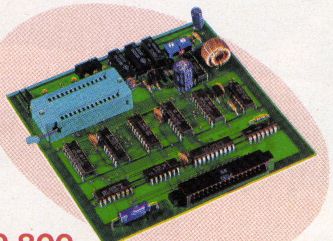
8ビット、4チャンネル、ANPORT命令サポート、低価格A/Dコンバータボードです。
BASICで簡単に使えます。



¥12,800

PIAボード ADA-7003

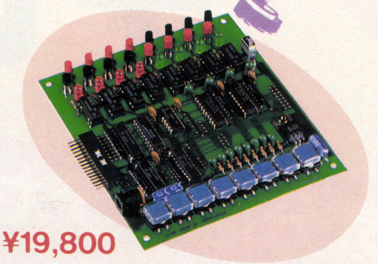
デジタル信号を入出力するためのボードです。PIA (MB-8874相当) 2個実装、割り込み可能、1ポートのみバッファ付き



¥19,800

ROMライターボード ADA-7004

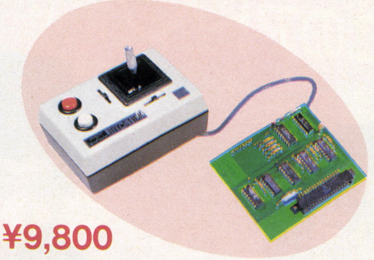
EPROM書込み器です。FM-7オプションスロットに挿入するだけで、即使用可能です。 専用ソフトウェアが付属しています。(カセットテープ)
対応EPROM:2716/2732/2764



¥19,800

接点入出力ボード ADA-7005

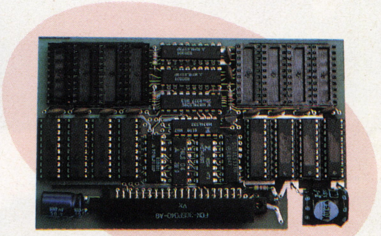
ADA-7003(PIAボード)と接続することにより、目で見るペリフェラル学習や種々なコントロールができる (わかりやすいマニアル付)
※PIAボード接続ケーブル (¥2,500)



¥9,800

JOYSTICK ADA-7

スティック+アナログボード+ゲームソフト (ペンギンダンス)
ADA-7002とジョイスティックの組合せです
X-Y2方向、スイッチ2個



¥19,800

C-MOS RAMボード ADA-7006

新製品

三層基板 (高密度実装)

- I/Oアドレスマップド接続のC-MOS RAMボード です。
- BASICプログラム、データ等を記憶させる外部記憶エリアとして使用可能です。
- 32KBの容量で、バッテリーにて停電保証しています。(16KB実装)
- 保証時間:200時間(32KB実装時) データの入出力は、BASIC又はマシン語の命令による。 (サンプル プログラム付)

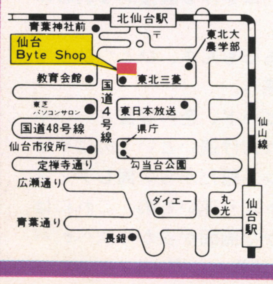
近日発売

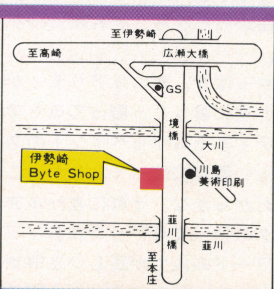
ADA-7007(RS-232Cボード)
COM 0:COM 1:COM 2:にて使用可能

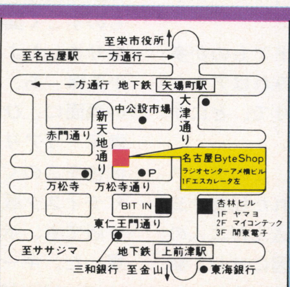
ADA-7008(FM-7用マザーボード)
32pinコネクター、4スロット
40pinコネクター、1スロット
を実装

あなたの街の Byte Shop

- 仙台バイトショップ 宮城県仙台市提通雨宮町3-18 〒980 ☎0222-33-0256
- 伊勢崎バイトショップ 群馬県伊勢崎市今井町755 〒372 ☎0270-23-2302
- 岡谷バイトショップ 長野県岡谷市幸町6-11(五十川ビル) ☎02662-3-1075
- バイトショップ プラスワン 東京都千代田区外神田1-8-11 ☎03-257-6337
- 関東バイトショップ 東京都千代田区外神田1-15-16(秋葉原ラジオ会館4F) ☎03-257-6341
- バイトショップ コーヨー 東京都千代田区外神田1-15-16(秋葉原ラジオ会館7F) ☎03-255-6504
- 名古屋バイトショップ 愛知県名古屋市中区大須3-30-86(ラジオセンターアメ横共同ビル) ☎052-263-1629
- 福岡バイトショップ 福岡県福岡市博多区博多駅前2-13-23(扇寿ビル) ☎092-474-5778
- バイトショップ管理部 東京都千代田区神田須田町2-23-1(芝崎ビル5F) ☎03-257-6308







Byte Shop サポート部
☎03-257-6309
〒101 東京都千代田区神田須田町 2-23-1芝崎ビル5F

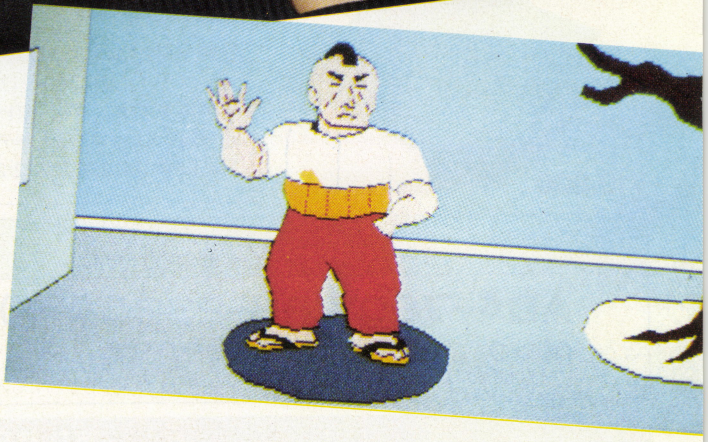
Hudson soft

デゼニランド

気分は、デゼニランド
このアドベンチャーゲームは、ホンモノです!!

竹部隆司(ハドソン・ソフトエンジニア)

1週間や2週間でGAME OVERになるんだったら、アドベンチャーの意味がないんですよ。ストーリーと絵の面白さを充分楽しんでもらいたいですね。



竹部 ストーリー・アドベンチャーというのは、要するにキミが主役になって、途中幾多の困難に見舞われながらも、ある与えられた目的を達成するという所、基本的なストーリーを、マイコン用にコーティングしたもの。

で、普通この種のゲームにはオカルトハウス的なインドアものが多いけど、「デゼニランド(出銭ランド)」の場合は、言ってみればアウトドア・アドベンチャーです。途中海あり、川あり、空ありで、とにかく広がりのあるゲーム展開が売りもの。

舞台は突然ディズニーランド（千葉県）にワープして、このディズニーランドをパロった五つのパビリオンのどこかに隠された秘宝「三月磨臼(ミツキマウス)」を探し当てるべく、まずはデゼニランドに乗り込んだ二人の少年に、デゼニランド・ショックを語ってもらいます。因にオレ、制作者の竹部です。

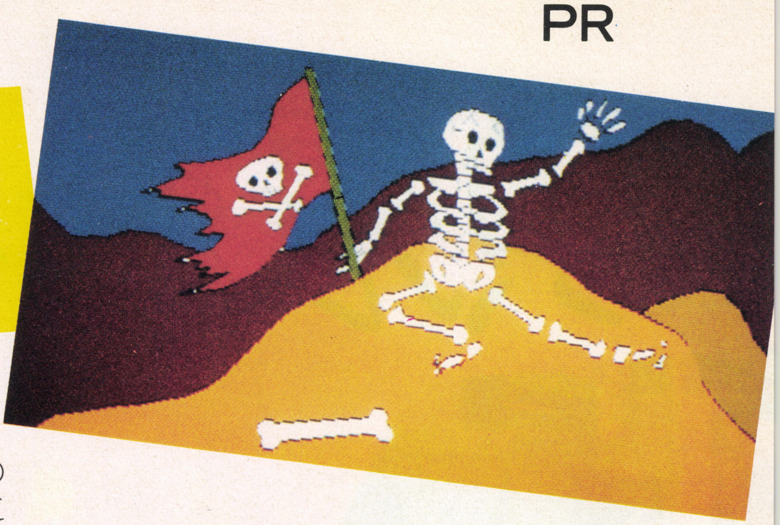
小久保 とにかく「ワー速い、きれい!!」がボク達の一致した感想ですね。それと、ナンとなく本物のディズニーランドへ行った気になる、というか、あ〜行ってみたいとゆ〜よ〜な、逆に言うと実際に行ったことのある人がプレイすると楽しさ倍増といった感じですね。

竹部 実際に現場を取材しましたからね。というよりは、行ってみてものすごく面白かったから、それがこのゲームを作る直接的な刺激になりましたよ。

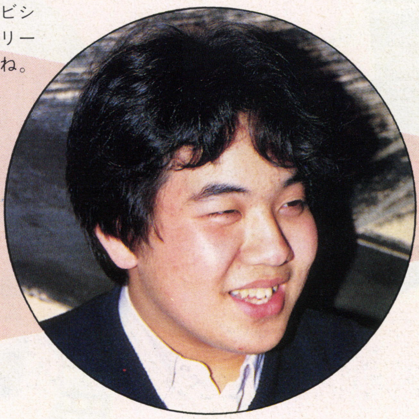
小久保 あ、やっぱり。実にリアリティがあるもんネ、ひとつひとつの場面に。ひと口に言うと、凝ってるって感じ。

君塚 リアリティとそれからペイントのスピード、色の豊富さ……中間色をこのくらい豊富に使って、しかもこれだけ速いというのはやはりディスク版だからかな。それにオールマシン語だから。同じディスク版でも、ボクの持ってるアップルのソフトより速いもの。これには驚いた。キーワードの入力が正しいと鮮やかな画面が次から次へ、パッパッパッパッだもんね。しかもそれが全部で114枚……!!

おもしろ体験記



君塚泰宏(都内在住、高校3年)
カラーの豊富さと、ペイントのスピードにただアゼン。全ての画面に意味があり、あらゆるキーワードが最後になってビシッとつながってくるストーリー展開の確かさもスゴイですね。



小久保秀次朗(都内在住、高校1年)
絵の鮮やかさと枚数もスゴイけど、にも増してキーワードの設定の巧みさに感心しました。いきなりグッグッと引き寄せ、最後まで緊張感を持たせるストーリーの確かさは、まるで推理小説のような迫力があります。あきさせません。

小久保　ただ入力の仕方が今までのアドベンチャーとはちょっと違う感じがする。英語入力でありながら、文法はまるで無視してるみたいな。

君塚　だから、さ‼変に英語力が無くとも楽しめるじゃない。ボクなんか独語専攻だから、単語だけで"何をどうする"式の会話型入力方式は助かります。間違うと、エラー・メッセージが出てくるし、引手に文法に気をつかわずにナンとなくフィーリングで入力して行けるものネ。

小久保　だから、いっそのこと日本語入力にすれば……。

竹部　それはネ、日本語にすると入力文字がどうしても多くなるから、むしろ中学生並みの単語知識で、ポツンポツンと入力して行けばいいように工夫したつもりなんだ。

君塚　う～ん、それにしても疲れた。もし竹部さんのサジェスチョンがなければ、ひと月やふた月は優にかかりそうな。下手をすると、パビリオンの中をあっちグルグル、こっちグルグル。

小久保　永遠に出られそうもない。そんなパビリオン5つもあって、そのひとつひとつのパビリオンが、従来のアドベンチャーゲームの1本分はある。そんなモノホンの迫力を感じましたね。

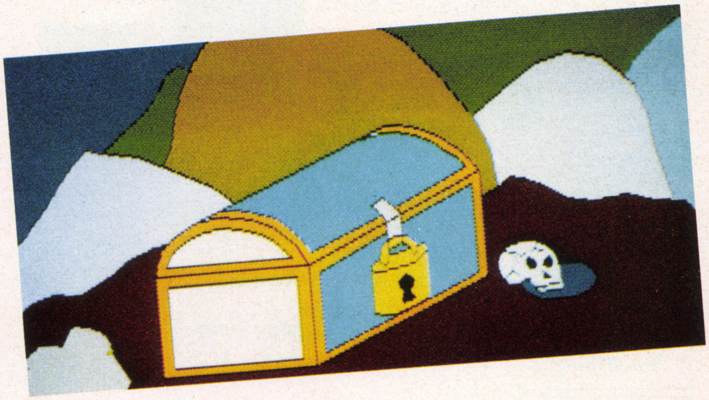
竹部　そこは、つまり制作者とユーザーの知恵比べというかね。それにハドソンとしても、今後のディスク時代に対応した揮心の第1作目ですから、力は入ります。デゼニランドはその意味で大サービスなんですよ。

君塚　ところで、この114枚の画面。これ全てに意味があるんですよね。最初の方は割と優しいヒントがもらえるからスイスイ行っちゃうけど、後になればヒントも少なくなるし、さてどうしようという時、それまでの画面やキーワードを覚えてないと立ち往生しちゃう。

小久保　ボクの場合は、途中で、あ、これはメモしておく必要があるなとネ。例えば、六つの店で買い物をする場面で、買い物リストを作るとか、全てのキーワードをメモっとくとか。そうしないと、これは解けない。

君塚　ウーン、いやはや何とも。全部で114画面の全てに後々必要になってくるキーワードが隠されているんだからね。それに、全部のキーワードが最後でビシッとつながってくるところに、ホンモノのアドベンチャーを感じます。ストーリーが生きてる‼

小久保　ウン、生きてる。いきなりグッグッと引き込み、最後まであきさせない。デキの良い推理小説並みの迫力です。ス・ゴ・イ‼

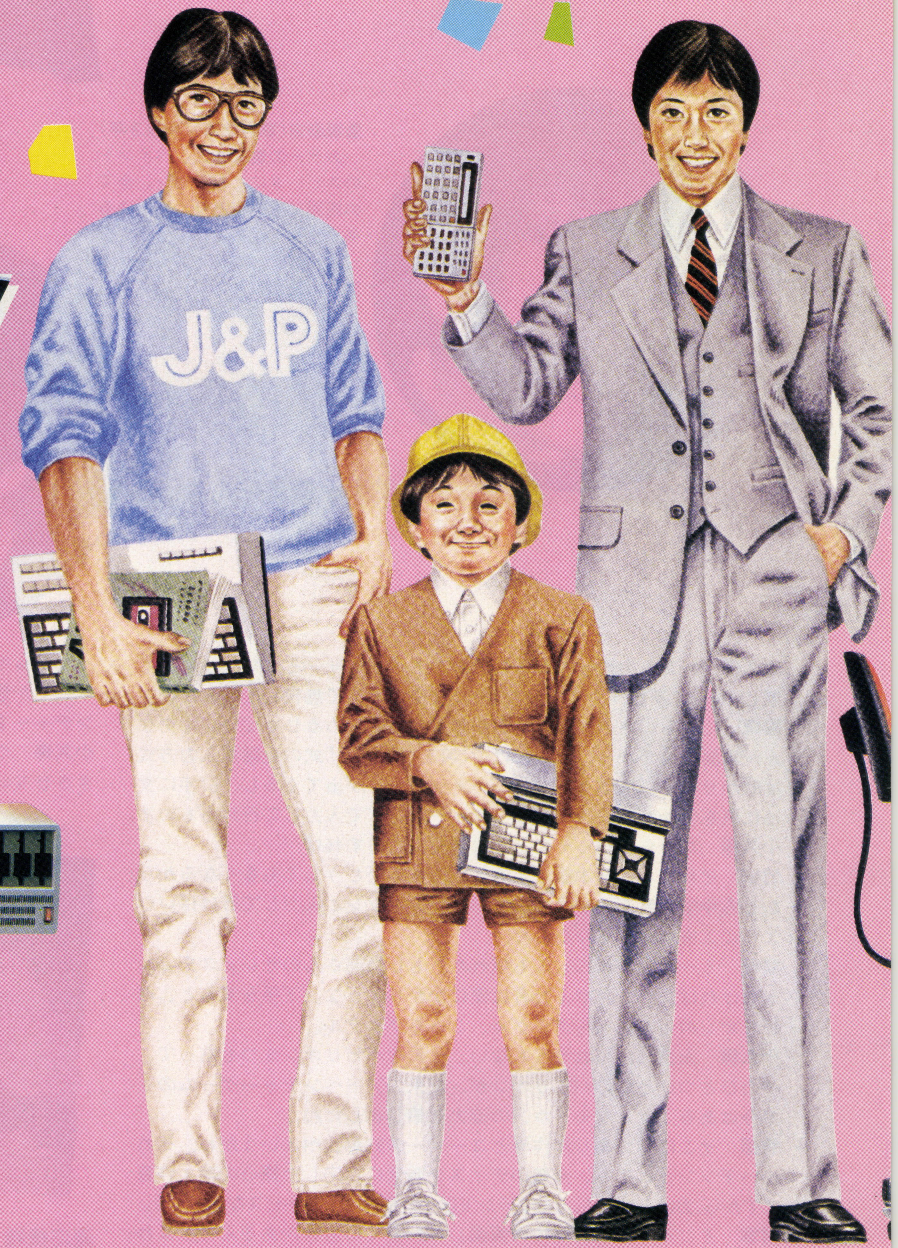


適応機種

PC-9801	5″FD	¥6,800	FM-7	T	¥4,800
PC-9801F	5″FD	¥6,800	FM-7	5″FD	¥6,800
PC-8801/mkII(共用)	5″FD	¥6,800	X1-D	3″FD	¥5,800
PC-8801/mkII(共用)	T	¥4,800	X1-C	T	¥4,800

HUDSON GROUP
HUDSON SOFT®
ハドソン札幌/〒062 札幌市豊平区平岸3条5丁目4番17号 コロナード平岸II 201 PHONE:011-821-1538
ハドソン仙台/〒980 宮城県仙台市宮町1丁目4番28号 PHONE:0222-65-7031
ハドソン金沢/〒920 石川県金沢市本町2丁目1番28号 PHONE:0762-23-1263
ハドソン東京/〒102 東京都千代田区麴町4丁目7番5号 麴町ロイヤルビル2F PHONE:03-234-4996
ハドソン大阪/〒542 大阪市南区南船場4丁目2番18号 佐野屋橋ビル5F PHONE:06-251-1945
ハドソンUSA/2063 CENTER STREET BERKELEY CA 94704 TELEPHONE 415-845-1416

頑張れ！パソコン



ハードウェア
MSXからIBM5550まで

誰にでも使えるMSX仕様パソコンからハイグレードなIBM5550まで、
国内・海外の人気機種をビッグに品揃え。
あなたの目的に合った機種がご自由に
お選びいただけます。
また、ディスプレイ、プリンタ、フロッピィディスクなど
周辺機器を集中展示しております。

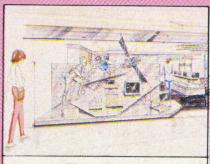
システムサポート

- **コンサルティング** OA化のご提案をします。（企業内教育・指導、OAシステム提案）
- **オーダーソフト** ご希望のソフトを作ります。（ビジネス用ソフト開発、特殊用途ソフト開発）
- **データサービス** データ入力のお手伝いをします。（データのインプット、ワープロサービス）
- **アフターサポート** オペレータの教育を引き受けます。（社内インストラクター教育、オペレータ教育）

全国をネットするパソコンの大型専門店J&Pチェーン。

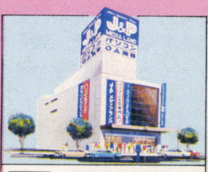
東京 渋谷店
東京都渋谷区道玄坂2丁目28番4号（〒150）
☎(03)496-4141

東京 町田店
東京都町田市森野1丁目39-16（〒194）
☎(0427)23-1313



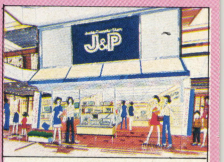
東京 八王子店
東京都八王子市旭町1番1号八王子そごう7F（〒192）
☎(0426)26-4141

大阪日本橋 テクノランド
大阪市浪速区日本橋5丁目6番7号（〒556）
☎(06)644-1413



大阪日本橋 メディアランド
大阪市浪速区日本橋5丁目9番11号（〒556）
☎(06)644-1613

大阪 くずは店
枚方市楠葉並木町2丁目2－2（〒573）
☎(0720)56-8181



大阪 千里中央店
豊中市新千里東町1-3-204 千里サンタウン3F（〒565）
☎(06)834-4141

一年生。

心はずむ春。進入学、就職と未知の世界に船出する君たち。まずは、おめでとう。
でも、浮かれてばかりじゃいけない。何か目標を決めてチャレンジしてみよう。
この機会に"自分のパソコン"を持って、本格的に取り組んでみるのもいいことだ。
J&Pでは、マニアはもちろん、初心者だって面倒見ちゃう。
君のパソコンライフ、応援するぞ。

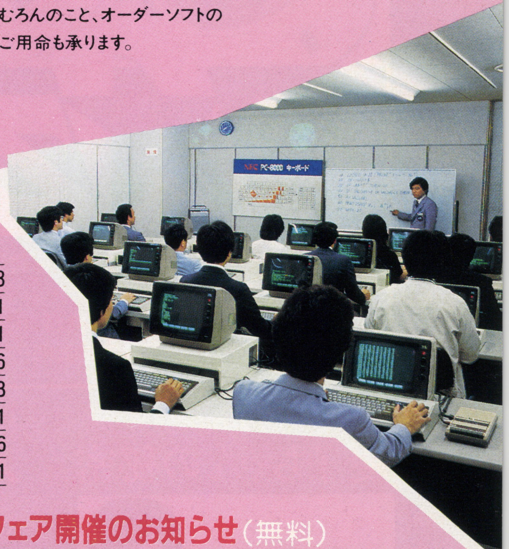
ソフトウェア

全国から集めた優秀なソフトは5,000種類。
お求めやすく機種別に分けた展示が好評です。
ビジネス用、ホビー用のパッケージソフトは
むろんのこと、オーダーソフトの
ご用命も承ります。

セミナー

初めての方でも、マニアの方にでも
分かりやすく楽しいカリキュラムの
各講座をご用意いたしております。
お問い合わせ、お申し込みは下記の
パソコン教室までお電話でどうぞ。

渋谷店	教室	(03) 496-4148
八王子店	教室	(0426)26-4141
町田店	教室	(03) 352-3131
テクノランド	教室	(06) 644-5566
京都寺町店	教室	(075)341-3573
くずは店	教室	(0720)56-8181
四日市店	教室	(0593)54-3366
津店	教室	(0592)26-0111

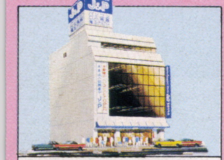


話題のビジネスソフトフェア開催のお知らせ（無料）

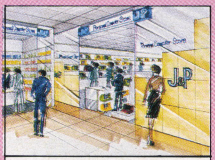
X-Yプロッタで漢字も描ける画期的CADソフト アポロテクニカ **漢字CADエース・フェア** 〈PC−9801(F)用・58,000円〉		20項目高速検索が可能な漢字顧客管理ソフト アポロテクニカ **漢客フェア** 〈PC−9801(F)用・48,000円〉	
実施日	会場	実施日	会場
3月24日㊏・25日㊐	テクノランド	3月24日㊏・25日㊐	メディアランド
3月31日㊏・4月1日㊐	町田店	3月31日㊏・4月1日㊐	渋谷店
4月7日㊏・8日㊐	京都寺町店	4月7日㊏・8日㊐	テクノランド

■時間はいずれも、1回目PM2:00〜3:00、2回目PM3:30〜4:30

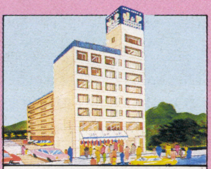
東京 — 大阪 — 京都 — 名古屋 — 三重 — 新潟



京都 **京都寺町店**
京都市下京区寺町通仏光寺下ル恵美須之町549(〒600)
☎(075)341-3571



名古屋 **栄ノバ店**
名古屋市中区栄3丁目4-5 SAKAE NOVA6F (〒460)
☎(052)261-9201



三重 **四日市店**
四日市市鵜の森2−1−19(〒510)
☎(0593)54-3366

三重 **津店**
津市丸之内31−20(〒514)
☎(0592)26-0111

新潟 **新潟店**
新潟市東大通1-3-10三井生命ビル1F(〒950)
☎(0252)43-6671

Personal Computer Store
J&P

CHAMPION SOFT

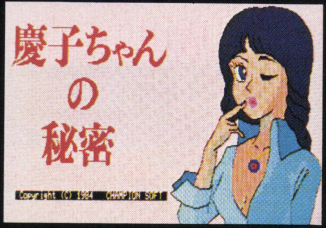
銀河戦争 **Aizma**

西暦2100年。200光年の彼方を舞台に、3D高速リアルタイム＋ロールプレイングゲーム。
FM-7, PC-8801・mkII各機種順次発売!! （両機種とも）
テープ版**4,200円**・ディスク版（5FD）**6,200円**（ペーパークラフト付）

アドベンチャー・シミュレーションゲーム
カタストロフィー

アメリカ国家安全保障基地レーダーにキャッチされたUFOを追って君は、地球を救う為に立ちあがった!! 分割ロード本格アドベンチャー。FM-7, PC-8801・mkII（各機種とも）
テープ版（3本組）**5,800円**　ディスク版（5FD）**7,800円**

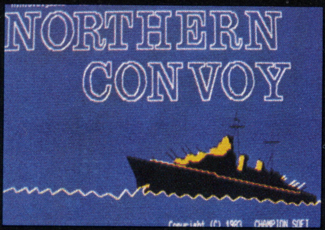
サスペンス・ラブアドベンチャー
慶子ちゃんの秘密



	テープ版	ディスク版
FM-7	○	○ 5D
PC-8801・mkII	○	○ 5D
PC6001・mkII	○	
PC-9801/F		○ 5D 5DD

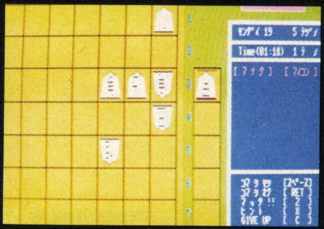
テープ版各**3,800円**　ディスク版各**4,800円**

シミューレーション・ウォーゲーム
ノーザンコンボイ



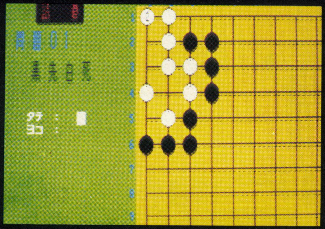
戦艦はムルマンスクをめざす。
FM-7,
PC-8801・mkII,
PC-8001mkII
X1シリーズ
各**3,800円**

将棋の決定版
詰将棋



「本物を」の声に応えて
●FM-7
●PC-8801・mkII
〈各機種とも順次発売〉
テープ版　各**3,800円**
ディスク版　各**5,800円**

新、囲碁トレーナー
詰碁



碁力、パワーアップ
●FM-7
●PC-8801・mkII
〈各機種とも順次発売〉
テープ版　各**3,800円**
ディスク版　各**5,800円**

⊙お求めは、お近くのマイコンショップまたは、チャンピオンソフト販売㈱通販部宛に品名と機種、氏名、住所、電話番号を記入の上、現金書留にてお送りください。（送料は無料サービス）

販売代理店募集中

制作 株式会社チャンピオンソフト
〒530 大阪市北区西天満6丁目1番12号瑞穂ビル
TEL(06)365−9900　ファックス(06)361−1176

発売元 チャンピオンソフト販売株式会社
本社・通販部 〒657 神戸市灘区篠原北町1丁目5番13号
TEL(078)801-2727　ファックス(078)882-0555
東京営業所 〒111 東京都台東区蔵前4丁目33-7山愛ビル
TEL(03)866—8447　ファックス(03)851-4880

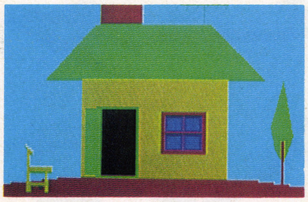
絶賛発売中

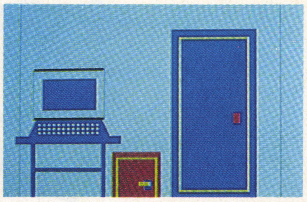
クリスタルソフト ADVENTURE SERIES

話題独占

難易度…★

名探偵登場!





「ちえっ、生まれつきのドジのおかげで、また会社をクビになっちまった。」僕はつぶやいた。「何か僕にもできる仕事はないだろうか。」そう思いながらぼんやり歩いているうちに『クリスタル探偵事務所』という看板を見つけた。「探偵かおもしろそうだな、一度やってみたかったんだ」「こうなったら何でもやってしまおう。」とやけくそで戸をたたいた。……というわけで、あなたは新米の探偵となりましたが、あなたは探偵として成功できるでしょうか?

STORY

FM-7 ¥4,500カセットバージョン

難易度…★★

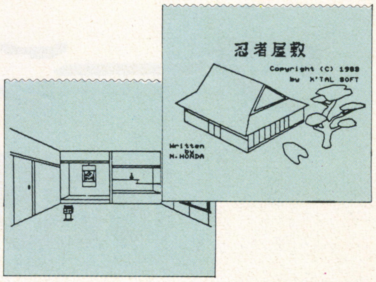
忍者屋敷

からくりアドベンチャーゲーム

FM-7 ¥4,200カセットバージョン

時は戦国時代、我が藩の重要な秘密を記した密書が敵の忍者によって奪われた。あなたに"我が藩の名誉にかけてもその密書を奪い返せ"という命が下った。これを取り返さねばあなたの命はない。あなたは仲間が敵の忍者たちを誘い出しているうちに屋敷に潜入し、密書を探しあてて奪い返さねばならない。だが行く手にはいろいろな仕掛けが待ち伏せている。無事成功を祈る。

STORY



難易度…★★★　アースバウンド

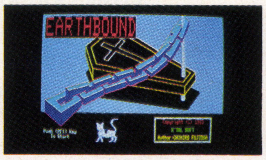
EARTH BOUND

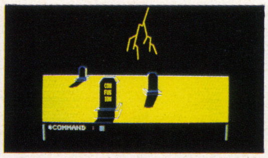
今までにないハイセンスなアドベンチャーゲーム

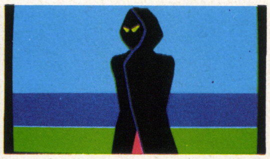
3分割ロードプログラム

大好評

FM-7 ¥4,500 カセットバージョン







ある朝、目を覚ますと、あなたは、自分がいったい何者なのか、そして、今まで何をしていたのか、何も思い出せない。たったひとつ思い出せるのは、レディ・ジューンという人の名前だけ。そこで、あなたはあなたが目を覚ましたコウマス・チュアという村を出て人の噂を頼りに、とうとうこのラ・テールの村の入り口にやってきた。ここには、ジューンという不思議な力を持つ女が住んでいるらしいのだ。とにかく、今あなたにできることは、この村の中で、レディ・ジューンを捜し出すか、それとも何か手掛りになるものを見つけることだけなのだ。

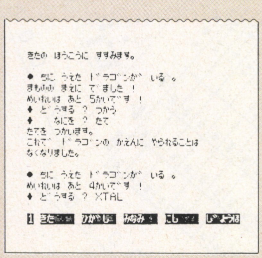
STORY

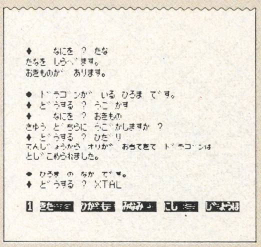
聖なる剣

日本語処理の文字によるスーパーハイレベルアドベンチャーゲーム

ある日、道ばたに一枚の立て札を見つけた。『東の山のはて、ポー山のどこかに住む悪い魔王にさらわれたかわいそうなお姫さまを救ってくれた人には何でも欲しいものをあげる。……王さま』うわさでは、魔王の住み家にはたくさんの怪物（吸血コウモリ・ドラゴン・ひとつ目巨人・へび女etc）が飼われていて、今まで救出しようとした人は全てこいつらの胃袋の中とか。勇気をふりしぼって魔王の住み家の入り口にたどりついたあなた。剣も盾も持っていないあなたに果してお姫さまを救出できるか。

STORY





難易度…★★★★　**FM-7・8 ¥4,200 カセットバージョン**

クリスタルソフト販売代理店

東京●㈱日本ソフトバンク	☎(03) 263-3690
大阪●㈱日本ソフトバンク	☎(06) 644-0191
東京●㈱ニデコ・ソフトセンター	☎(03) 253-0761
名古屋●大江㈱	☎(052)851-7251
福岡●㈱メディア	☎(092)474-9065

XTALSOFT
Personal Computer Software

●ロゴマークが新しくなりました

——全国無料即日発送——

通信販売もいたしております。ハガキ大の紙に品名、定価、住所氏名、年令をお書きになって、現金書留か、小為替で下記までお送りください。送料は、サービスいたします。

クリスタルソフト株式会社
〒533大阪市東淀川区豊新3-8-11 いほりビル2階　TEL(06)326-8150

●忍者屋敷に一部エラーのでるものがあります。該当品をお持ちの方は、直接当社までご連絡ください。

漢字データベース

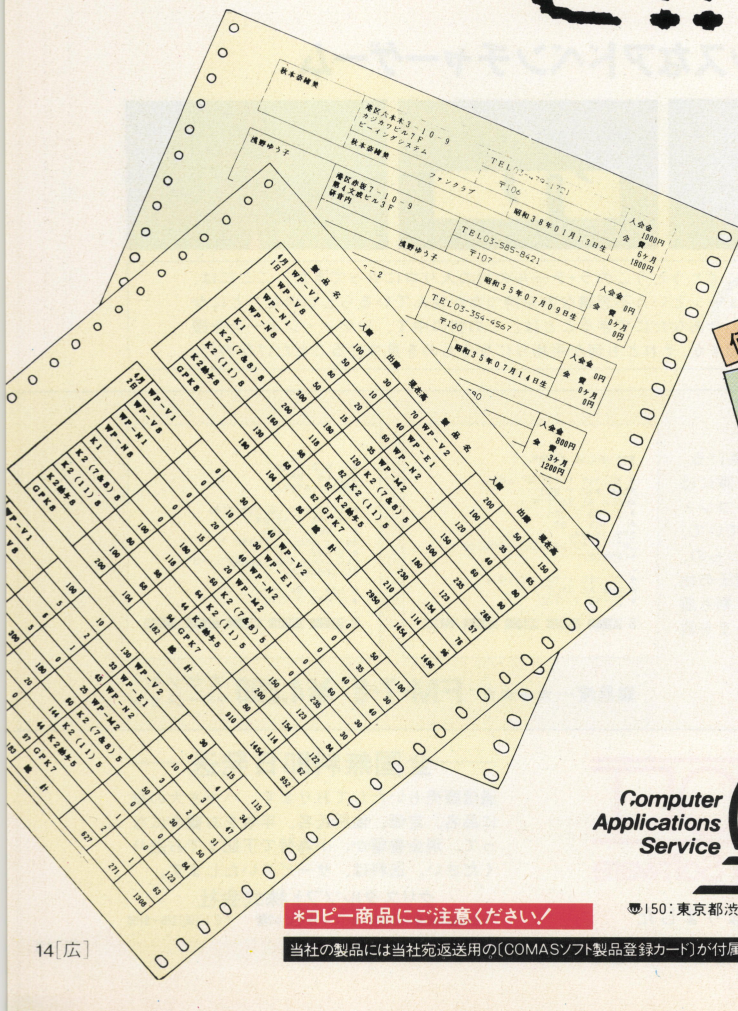
COMAS K2シリーズ

データ入力はワープロ感覚で!!

データベースは入力したデータの処理能力が第一……
確かにそのとおりですが、入力するデータが何百件・何千件とあるとき、データ入力の容易さは重要な問題です。
これはたとえば、どんなに記入項目が少ないとしても、1000枚のカードにデータを手書きで記入することを考えれば想像できます。
また、入力データの追加,削除,訂正等も容易でなければなりません。
パソコン・データベースを使ってみれば、「ワープロの入力機能が使えれば!!」と思うのは当然です。

COMAS-K2シリーズでは、熟語単位のかな漢字変換,ローマ字変換による入力機能があります。
内蔵辞書も約2万語と豊富ですし、当社WPシリーズの熟語ファイルを使用することもでき、K2Gではウレシイことに学習機能までもっています。
そのうえ,郵便番号入力による"地名変換",記号入力による"人名変換"といった、データベース用のユニークな入力機能もあり、挿入・削除・複写・短文登録等,その他の日本語処理機能も豊富です。

しかし、いくら入力が容易でも、データ処理がオソマツでは話になりませんが、その点も心配はありません。
COMAS-K2シリーズは、オールマシン語プログラミングによる高速処理と、多様な処理機能を誇るデータベースシステムです。



COMAS-K2G, K2Sは、従来の"COMAS-K2"を大幅に機能アップしたものです。
従来のK2は、対応プリンターを増やして、"COMAS-K2E"として販売いたします。

対応プリンター

MB27401, MB27402, MB27403, MB27405, MB27406, MB27407, MB27410
MP-80, MP-80TYPEII, MP-80TYPEIII, MP-100, MP-80K, MP-130K
DPX-510F, DPX-510J, radix10, radix15
GP-550E, KP-3000, RITEMAN

価格

COMAS-K1(FM-7&8用のみ)
5インチ版……………¥33,000

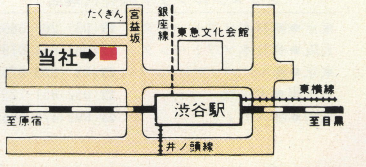
COMAS-K2E
5インチ版……………¥55,000
8インチ版……………¥85,000
FM-11 2HD用………¥85,000

COMAS-KSG
5インチ版……………¥85,000
8インチ版……………¥120,000
FM-11 2HD用………¥120,000

COMAS-K2S
5インチ版……………¥99,800
8インチ版……………¥135,000
FM-11 2HD用………¥135,000

※K1を除いては、FM-7&8用とFM-11用があり、FM-7&8用はFM-7とFM-8で共用できます。

Computer Applications Service
株式会社コマス
パソコン事業部
〒150:東京都渋谷区渋谷1-14-6(平野屋ビル2F)☎(03)407-8893(代)



*コピー商品にご注意ください!

当社の製品には当社宛返送用の〔COMASソフト製品登録カード〕が付属していますので、製品お買い上げの節には、かならずこの点をご確認のうえ、当社宛ご送付いただくようお願い致します。

「楽しさ満点。思わず興奮。」

サンダーフォース

スクロールゲーム

シールドに被い隠されその姿を容易に現わさないといわれるORN(オーン)軍秘密要塞「ダイラデイサー」。これを破壊すべく、惑星連邦の最新爆撃宇宙艇「ファイアー・レオ」は、小惑星帯の敵基地に突入。しかし、ORN軍の迎撃は熾烈を極めた。ダイラデイザーを破壊するにはまずシールドベースを発見し、これを破壊しなくてはならない。これを破壊すると眼下に超巨大要塞「ダイラデイザー」が……。

美しい画面にくり広げられる熾烈な戦い。縦横にスクロールする画面、迫力ある効果音、臨場感があなたを未知の戦闘空域へ誘います。

★FM-7、PC-8801、8801MKII、X-1C
X-1(G-RAM)用　　各￥4,800

プロも絶賛 ゴルフ・アイランド

思わず楽しくなる超リアルな演出のグラフィックとサウンド。

単にボールを打つだけではありません。ボールが木にあたるとちゃんとはね返り、しかも木の葉も落ちます。それに池に落ちると水しぶきが、バンカーに落ちると破けむりが上ります。また同じホールでも、木、池、バンカーの出る位置が毎回違います。こんなゴルフ、他にあるかな？

★FM-7、X1(G-RAM)、PC-8801、PC-8001MKII、PC-6001MKII、MZ-2000(HU-BASIC、G-RAM1、2、3)　　各機種￥4,000

★別売り拡張コース、(1～4)で更に多くのコースが回れます。　各￥2,000

➡「ゴルフアイランドスペシャル」(基本コース＋拡張コース4の2本セット)も大好評発売中!!

各機種￥5,800

DATE

四人麻雀

狙いは夢の"九連宝燈"

完璧に本物と同じ！四人麻雀。リーチ、ポン、チー、裏ドラ一発ありで東・南まわしの半荘ジメ。

速いスピードと最高にきれいなグラフィック。真剣勝負の醍醐味を充分味わってキミも雀狂にならないか。

★FM-7、X1、PC-8801、PC-8001MKII、PC-6001MKII、MZ-2000 / 2200(dB-BASIC、G-RAM1、2、3)

各機種￥4,000

まさに本物の迫力！

スターフリート/B

DATE

3次元宇宙空間で繰り広げられる驚異のシミュレーションゲーム。リアルな動き、要求される冷静な状況判断力。そして決断力面白さがナミじゃない。

★FM-7/8、X1(G-RAM)、PC-8801、PC-8001MKII、PC-6001MKII、MZ-2000(G-RAM1)

各機種￥4,500

君はもう体験したか？3次元ならではのトリップ感。

●5インチ両面倍密度説明書付　●各￥10,000

Super T-DOS7　Super T-DOS8

FM-7用：30530Bytes　FM-8用：30358Bytes

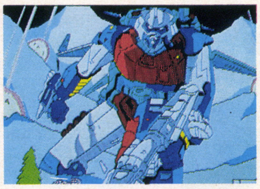
①ROM-BASICと同じフリーエリア(30530バイト：FM-7, 30358バイト：FM-8)を保ちつつ，DISKに対して、LOAD/SAVEが可能となりLOADエラーから開放されます。
②ROM-BASICの時の命令は全て使用できます。
③1枚のディスケットには，最高で152のプログラムが収納可。
④PF-KEYの設定をBOOT時に行なえます。
⑤FILESの表示では，縦横に表示されますので見やすい。
⑥命令に変数が使えます。
(例)A$＝"0 :TEST":RUN""＋A$
⑦BASICのTEXTのSAVEやASCIIでのSAVE, MERGE, LOAD又マシン語のLOAD，SAVEその他多数の命令が使えます。
⑧データをカセットにSAVEしていたものが，そのまま又はごく少しの修正でDISKに対して，SAVE，LOADできます。
(例)顧客管理，在庫管理，スケジュール管理，成績処理等。
⑨テープ版で番地不明のマシン語がディスクへ対してSAVE，LOADできます。

ディスクでROM-BASICと同じフリー・エリアが使えます。

グラフィック・ツールセット

2種のグラフィック・ツールで、精密なグラフィック画面が、さらに簡単に描けるようになりました。

FM-7/8用2本セット　￥4,500



(1)ライン・エディター
●多機能エディターで思い通りの絵が簡単に描けます。
●8原色を同時に異った中間色に変換できます。
●作成した絵をCONNECT、PAINT、LINE、SYMBOL等の文として、あるいはマシン語として出力します。
●BASICのテキストをデーターとしてロードし、改めてエディトする逆変換機能もあります。

(2)パターン・エディター
●100×50ドットの大きさまでのパターンが作成可能。
●1ドット毎の色つけができます。
●エディト時に直線、円、塗りつぶし等もできます。
●作成したパターンはデーター文としてファイルに出力されます。

㈱テクノソフト　AM10:00～PM6:00 日曜定休

〒857 長崎県佐世保市福石町4-14 ☎0956-33-5555

「写真は全てFM用です。」★お求めは、お近くのマイコンショップ、又は当社へ現金書留にてお申し込み下さい(送料無料)

「普通のアドベンチャーゲームはもういらないよ」という人に、新発売。

ハミングバード〈別ベンチャー〉シリーズ 第1弾

骨の髄まで、面白い。

痛快！企業内外冒険活劇電子遊戯超大作

RECAPTURE

リキャプチャー〈奪還〉
富士通FM-8・7用 5インチディスク版
Humming Bird BE-TSU venture #1

値下げ断行
¥9,800

盗まれた新薬製造機密書類をとりかえせ！
製薬会社のダメ社員が、決死の奪還作戦を開始した。
楽しくも、ややこしいパロディ・アドベンチャー。
絶望的に、確実に、あなたの頭はショートする。

(犠牲者の一例)
実験的にこのゲームを
プレイしてもらった人の変わり果てた姿です。

このゲームを最後までプレイされた結果
発生するかもしれない、各種症状の一例

●肉体的変質●精神崩壊●社会的不適応性性格変容
●性的異常症状(性的不能を含む)●失恋・離婚
●自己破産●停学・退学・留年・浪人

その他あらゆる破滅的状況について、当方は一切の責任を負いかねます。
よおく覚悟を決めてお買い上げください。

(あなたのご幸運とご無事を祈ります。)

Humming Bird "BE-TSUventure" Series は、ハミングバード・アドベンチャーゲーム・シリーズのアナザーフェイス。通常のアドベンチャーゲームのワクにとらわれない自由な発想で、楽しい作品をリリースしてゆく予定です。ご期待ください。

"パームス"やったか、"アビス"はまだか。

Humming Bird Adventure Game Series Line-up

た、大変！フィアンセが海底人にさらわれちゃった。

#1 **"THE PALMS"** ザ・パームス
(富士通FM-8&7用 5インチディスク版)

旺文社パソコンゲームランキング 和製アドベンチャーゲーム部門 第3位

¥11,000

早く、早く！"魔法の水"を手に入れないと、姫が死んじゃう。

#2 **"THE KNIGHT OF WONDER LAND"** ザ・ナイト・オブ・ワンダーランド
(富士通FM-8・7用 5インチディスク版、NEC PC—9801用 5インチ両面or 8インチディスク版)

旺文社パソコンゲームランキング 和製アドベンチャーゲーム部門 第1位

¥11,000

〈Mission:ABYSS〉を遂行せよ。

#3 **"ABYSS"** アビス
(富士通FM-8・7用 5インチディスク版)

〈ヒントブック〉、出ました！ ¥700

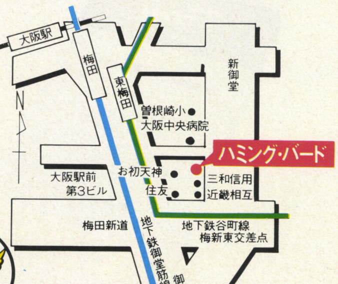
¥9,800

全国の有名パソコンショップで、好評発売中！

ハミングバードのショップも、よろしく。

Appleソフトの品ぞろえが、ダントツ。まあ一度のぞいてくださいな。 アドベンチャーゲームのスタッフたちがお待ちしてます。

(通信販売もしています。下記までお問い合わせください。)



Humming Bird Soft
㈱エム・エー・シー コンピュータ事業部
〒530 大阪市北区曽根崎2丁目2番15号 ☎06(315)0541(代)

PNシリーズ続々新登場!!

PN-101(FM-8用) 30,000円
PN-101(FM-11用) 40,000円
PN-102(FM-7用) 12,000円
(辞書メインテナンス、顧客ユーティリティ)

PACKET NETWORK SANWA BUSINESS

パソコン用日本語ワードプロセッサーがついに電子タイプライターを超えました。従来のワープロの処理階層をなくし、すべて同一画面でワンタッチ操作が可能なダイレクトコマンド方式、また、1ページの文書が一度に入力できる連続文書一括入力などを取り入れ、文書作成時間を今までの2分の1以下に短縮しました。これからのパソコンになくてはならないソフト、それがPN-101です。

未来に技術革新 開発元
SANWA BUSINESS 三和ビジネス株式会社
〒231 横浜市中区山下町75番地(荻野ビル8F)
TEL.(045)641-6401(代表)

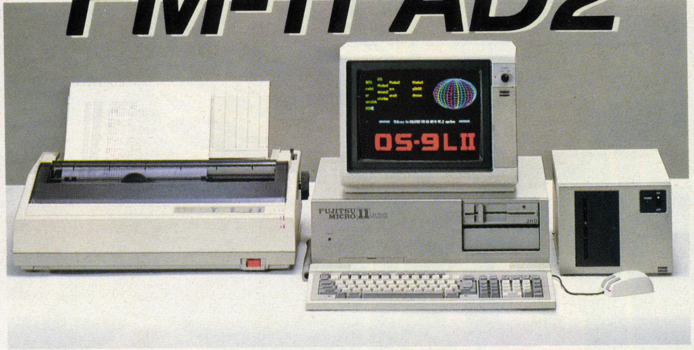
総販売元 岡田商事株式会社
オスコム事業部
〒105 東京都港区芝大門1-3-7
☎03(459)8217(代)
担当金子・嶋野
北海道支店/〒060 札幌市中央区北6条西13-1-15
☎011-241-2401 担当高橋
東北支店/〒983 仙台市原町1-1-86
☎0222-91-2391 担当斉藤
浜松支店/〒435 浜松市篠ヶ瀬町977
☎0534-64-8159 担当上原
名古屋支店/〒462 名古屋市北区辻本通3-14
☎052-914-5991 担当植田
大阪支店/〒564 吹田市南吹田5-13-19
☎06-386-0161 担当三木
広島営業所/〒733 広島市西区南観音7-4-12
☎0822-92-6479 担当沢重
高松営業所/〒760 高松市木太町2666
☎0878-34-7522~4 担当茂木
福岡支店/〒810 福岡市博多区那珂3-16-34
☎092-473-5331 担当渡辺

FM-11のニューフェース

西村 義孝

1MBの5インチFD内蔵

FM-11 BS　FM-11 AD2



去る2月15日，富士通よりFM-11 BSとFM-11 AD2が発表された。両機は，従来のFM-11の機能を拡充し，ビジネスユースなどに幅広く対応できる仕様となっている。基本的には，これまでのFM-11がベースとされているため，この稿では，FM-11 BS，AD2がどのように変わったかを紹介する。

FM-11 BS

FM-11BSは，メインCPUを6809ではなく，8088(8MHz)のみとしたCP/M-86マシンとなっている（サブ関係は，ほぼ変更なし）。主なスペックは，

- JIS第1水準の漢字ROM標準実装
- 1MBのミニフロッピ2基搭載
- 256KB（メインメモリ）標準実装
- CP/M-86 V1.1 Level 3標準添付
- F-BASIC86バンドル
- JWORD　標準添付
- マウス（オプション）
- MS-DOS Ver2.0（オプション）

である。以下それぞれ詳しく解説する。

1　1MBのミニフロッピ搭載(X2)

これまでのミニフロッピといえば，両面倍密（2D 320KB）が普通であったが，カセットテープの代わりならともかく，OSを動かして本格的に使いはじめると，320KBは少なすぎるといえる。とくにCP/M-86の言語プロセッサなどはかなり巨大で，コンパイラ，リンカ，ユーティリティなどを1枚のディスケットに入れてしまうことができないこともままある。いきおい，フロッピを入れ替えてコンパイルを行うわけだが，非常に面倒である。またビジネスで大量のデータを使う場合，「容量がないので動かない」では許されないだろう。そのため最近の16ビットパソコンでは，倍トラック（2DD 640KB）のものが多い（NECのPC-9801F，東芝のPASOPIA16，IBMの5550，カシオのFP-3000)。

ところが，FM-11 BS（AD2も）に搭載されているフロッピは2HD（両面**高密度**倍トラック）である。この5インチドライブは8インチドライブと電気的特性が同じで，ソフト的には8インチドライブと同じように使える。容量ももちろん8インチディスケットと同じ1MBを持つ（フォーマット時)。この2HD(電電公社仕様と呼ぶ場合もある）を採用しているパソコンには，カシオのFP-5000があるが，まだまだ少ない。主流となるのは2DDか2HDか興味深いところである。2HDとなって問題なのは，メディアの問題である。これまでの2Dディスケットは使えないため，2HD専用のディスケットを使わなければならない。もっとも問題なのは，メディア変換である。2HDのドライブでは従来の5インチ2Dのディスケットは読めない。そのため，これまで開発した2Dのプログラムを2HDに送ることができない。これを行うには，BS側に従来の5インチの2Dドライブを付けて転送するか，RS-232Cでシリアル転送するしかない。もっともこれからBSのソフトウェアを買われる方は2HDで供給されるため問題はない。また，8インチFDUはオプションのケーブルを接続することによって容易に増設できる。

2　256KB(メインメモリ)標準実装

FM-11 BSはCP/M-86マシンとして使われることが前提とされている。CP/M-86上で動く言語プロセッサ，データベース，アプリケーションソフトウェアは膨大な量があり，その質も大型コンピュータ用のものと劣らないものとなっている。それは大型コンピュータにさせていたような仕事をパソコンでも充分できるようになってきたことを意味するが，質が高くなったかわりに，プログラムサイズも8ビットの時代よりも，はるかに大きくなっている。たとえば大半のCコンパイラはTPAが最低128KB以上ないと動作しない。これまでの16ビットパソコンは大半が標準実装は128KBであり，TPAはそれより少ない(96KB程度)ため，多くのユーザはメモリを増設するか，96KB内でどうにか動く，限られたソフトウェアをがまんして使っているのが現状である。実用上はやはりTPAが128KB以上必要で，メインメモリ256KB標準装備は妥当であろう。また，FM-11 BSでは1MBまで拡張可能で，オブジェクトサイズの大きなアプリケーションにも充分耐え得る。またFM-11 BSのCP/M-86ではRAMディスクをサポートしているため，TPAを削ってディスクに割り当て，高速化を図ることもできる。

3　CP/M-86 V1.1 L3標準添付

FM-11 EXや8088カードにバンドルされていたCP/M-86を機能強化したCP/M-86 V1.1 Level 3が標準装備されている。

主な改良点は，

(1)辞書のサポート

(2)メモリディスクのサポート

(3)プリントスプーラのサポート

である。

(1)　辞書のサポート

ビジネスユースでは，漢字が入出力できることが必要条件となっている。ただ，漢字コード入力や1文字変換による入力は，ないよりはましという程度で，実際にビジネスで使うには，強力な「かな漢字変換」機能が必須である。これまでこれは，アプリケーションプログラムで行ってきたわけだが，やはりOSでサポートすべき機能であろう。NEC 9801F用のCP/M-86，MS-DOS，IBM5550用のMS-DOSなどではOSレベルで辞書を持ち熟語変換機能をサポートしている。

もちろんFM-11 BSのCP/M-86でも，OSレベルで国語辞書を持ち，ワードプロセッサ相当の交換が可能となっている。(約37,000語。熟語単位のカナ/ローマ字漢字変換入力)。またEDTDICコマンドで単語登録，削除，並びかえなど，辞書のメンテナンスができる。また辞書の空エリアがなくなった場合，EXTDICコマンドにより空スペースを取り，新辞書を作ることもできる。

(2)メモリディスクのサポート

FM-11 BSのCP/M-86では，メモリディスク（RAM DISK）をサポートしている。

アセンブルやコンパイル，リンクの時間の多くがディスクアクセスに費されているが，メモリを仮想ディスクに割り当てることにより，高速化が図れる。

BSのメモリディスクは，ドライブM:でサポートされる。割り当てるメモリ量は，

64KB，128KB，256KB，512KB

が指定でき，modsysコマンドで，Loaderにセットする。メインメモリの一部を仮想的にドライブに割り当てるので，TPAはその分減少する。

TPA=メインメモリーシステムサイズ−メモリディスクサイズ

FM-11 BSは標準では256KBであるが，メモリディスク容量としては実用上128KB程度は必要だろうから，メモリを増設したほうがよいだろう。

8088のようにメモリ空間が1MBもある場合，1MBの大半をユーザプログラムに使うという場合は少なく，メモリディスクに割り当てるのは効率がよい。

当然ながら，

PIP　M:TEST2.C=M:TEST.C

としても，まったくディスクアクセスの音はせず，あっという間に転送が終了する。

(3)プリントスプーラのサポート

FM-11 BSのCP/M-86では，プリントスプーラをサポートしている。プリントスプーラはFLEX，MS-DOS Ver2.0ではサポートされているが，CP/M-86はOSとしてはサポートしていない。プリントスプールは，バックグランドジョブとして，ファイルのプリントアウトを行う機能で，エディット中にファイルをプリントしたりできるため，プリンタにマシンを占有されてほかの仕事ができない，ということがなくなり，効率的に作業を進められる。これは，マルチタスクにほかならないが，CP/M-86はシングルタスクであるため，マルチタスクをシミュレートする形でスプールが実現されている。そのため若干の制約がある。プリントスプールを行うコマンドは，PRINTコマンドで，

PRINT　ファイル名 ↵

とする。この場合，**ドライブ名は省略できない**。プリントキューは10個あるため，ファイルを10個までスプーリング待ちとして指定可能である。

スプーリング中のファイルの**読み出し**は自由に行えるが，**書き込み**は動作が保障されない。スプーリングを中止するには，単に

PRINT OFF ↵

でよい。

これらの機能のほか，BSのCP/M-86では，

- オープニングメッセージの日本語化
- 漢字入力のモジュール化
- コンソール出力コードのデータ長の設定
- エラーメッセージによる画面破壊防止
- 不当割り込み発生時の処理改良
- ユーザプリンタハンドラのサポート
- 回線制御機能の強化
- 10MB，20MBハードディスクのサポート

などが，新しく盛り込まれている。

4　F-BASIC 86バンドル

CP/M-86の下で動作するF-BASIC 86が標準装備されている。

5　JWORD標準添付

FM-11 BSには，エーセルのJWORDがバンドルされている。JWORDは，PC-9801のMS-DOS版，CP/M-86版が有名である。

JWORDの特徴は以下のとおり。

- 文節単位のかな漢字変換
  モウシアゲマス。→申し上げます。
  など。また，学習機能もある。
- 約65,000語の単語辞書
- 約7,000語の漢字辞書
- 約10,000語以上のユーザ用単語および定型句の登録（1単語は38字以内でその読みはカナ文字換算で12文字以内）
  バンク→株式会社日本ソフトバンク出版部Oh!FM編集室
- かな漢字変換の入力は，カタカナ，ローマ字，カナローマ字混在の3方式
- 強力な作表，演算機能
  カーソルキーによる作表，四則演算
- アセンブラ記述で超高速動作

6　マウス（オプション）

最近，異常な関心を寄せられているマウスが接続可能である（**写真**参照）。

マウスは　ポインティングデバイスの1つで，「CRTのグラフィックカーソルを動かし，各種メニュー図形（アイコンと呼ばれる）を指してメニューを選ぶ」といった使い方が一般的である。現在，マウスが付かないと16ビットパソコンじゃない，といった感があるが，マウスを充分生かしきったソフトウェアはあまり見受けない（NEC PC-100のJSWORD，PC-9801用CANDY（ASCII）などしかない)。マウスを生かすも殺すもソフトしだいと思うのだが，FM-11のマウス用ソフトウェアに期待したい。

7　MS-DOS Ver2.0（オプション）

MS-DOSはCP/Mを強化したOSで，今やCP/M-86とともに8086用OSの標準となりつつある。MS-DOSもVer 1.25からVer 2.0となり，階層構造のディレクトリ，I/O Redirection Pipe機能，スプーリング機能，強力なバッチコマンドが備わり，Xenix（Unix）的な使い方ができるようになった。ソフトウェアも，CP/M-86のものは大半MS-DOSにもある。

CP/Mとの互換性を最大限に考慮されたCP/M-86が主流となるか，UNIX的な

表1　FM-11 BSのCP/M-86上開発ツール

品　名	内　容	価　格	販売時期	販売元
C BASIC-86	ビジネス用BASICインタプリタ言語，漢字入力可能	140,000円	59/2	① ②
C BASIC COMPILER 86	C BASIC用のコンパイラ	210,000円	59/2	① ②
PL/I-86	ANSIサブセットG規格に準拠のPL/Iコンパイラ	263,000円	59/2	① ②
Pascal/MT+86	PASCALコンパイラ	200,000円	59/2	① ②
C-86	高級C言語 86版	250,000円	59/3	②
SSS FORTRAN-86	CP/M-86用FORTRANコンパイラ	212,500円	59/2	① ②
漢字LEVEL II COBOL	LEVEL II COBOLの漢字版	198,000円	59/2	③
FORMS-2	LEVEL II-COBOL用の簡易COBOLプログラムジェネレータ	40,000円	59/2	③
ANIMATOR	LEVEL II-COBOL専用デバッキングツール	58,000円	59/2	③
PASCAL/MT+86	PASCALコンパイラ	未　定	59/3	③
C-86	Cコンパイラ	未　定	59/3	③
Word Master 漢字版	スクリーンベースの高級編集機能をもったソフトウェア	34,000円	59/2	③
FILTRAN-86	IBM↔CP/Mのディスクフォーマット変換ソフトウェア	80,000円	59/2	① ②
SID-86	アセンブラレベルでの開発を支援するシンボリック・デバッガ	70,000円	59/2	① ②
Programmer's Utility	リロケータブルアセンブラ，リンケージエディタ，ライブラリアンなど	70,000円	59/2	①

①(株)マイクロソフトウェアアソシエイツ　②ソフトウェアインターナショナル(株)　③富士通(株)

使い方のできるMS-DOSが主流になるか興味深いところである。

このようにFM-11 BSは大容量，強力な日本語機能を持つため，その名のとおりビジネスユースにはパワフルな実力を示すマシンである。**表1**に発売あるいは発売予定の開発ツール言語を示す。

FM-11 AD2

FM-11 BSが，8088のビジネスマシンとすれば，AD2は6809によるビジネスマシンといえるだろう。ハード的には1MBフロッピ（×1）となり，漢字ROMが標準実装となった以外，ほとんど変更はない。スペックは，以下のとおり。

- OS-9 Level 2　標準添付
- 1MBのミニフロッピ標準装備（×1）
- F-BASIC V5.0 標準添付
- 日本語ワードプロセッサを添付
- マウス（オプション）

1 OS-9Level 2V1.2J標準添付

FM-11 AD2には，あのOS-9 Level 2（最大1Mバイトのメモリ空間管理）がなんと**標準添付**されている。リアルタイム・マルチユーザ，マルチタスク，マルチウィンドウに加え，このOS-9は日本語機能がサポートされている。

JIS第1水準の単漢字辞書が付いており，カナ入力，ローマ字入力，混合入力のいずれでも漢字変換が可能。また，ローマ字入力時はリアルタイムでかな変換される。辞書ファイルは学習機能付きで，ユーザは自分専用の辞書ファイル指定ができる。

またOS-9には当然ながら，

- BASIC09
- マクロテキストエディタ
- インタラクティブアセンブラ
- インタラクティブデバッガ

は付いてくる。OS-9で気になるのは，市販ソフトウェアが少ないことである。言語プロセッサ，データベース，アプリケーションなど，数えるほどしかリリースされていない。これは，OS-9自体が普及してくれば，徐々に解決されることであろうが，今FM-11とOS-9があって「何ができるんだ？」と聞かれても，「BASIC09が動きます」としか答えられないのが現状である。OS-9は強力なOSで，8ビットのUNIXと呼ばれるくらいであるが，CP／Mなどに比べ，まだまだソフトウェアが少ない。

FM-11，7のOS-9がリリースされて以来，かなりユーザが多くなったそうである。1日も早くアプリケーションソフトウェアが充実してほしいものだ。**表2**に，発売あるいは発売予定のソフトを示す。

表2　FM-11 AD2のOS-9上ソフト

品名	内容	価格	販売時期	販売元
●開発ツール				
Cコンパイラ	標準Cコンパイラ	160,000円	59／2	④
PASCAL	標準PASCAL準拠	160,000円	59／2	④
CIS COBOL	ANSI-74準拠	300,000円	59／2	④
FORMS-2	COBOLのプログラムジュネレータ	88,000円	59／2	④
FORTRAN77	FORTRAN	未定	59／5	④
OS-9スクリーン	フルスクリーンエディタ	未定	59／3	④
LISTユーティリティ	F-BASIC，FLEX，CP／M上のファイルをOS-9上に変換	14,800円	59／2	⑤
ファイル転送プログラム(ROMライタ用)	RS-232CによりROMライタにデータを送る	50,000円	59／2	⑤
●データベース				
SAGE	データベースソフトウェア	未定	59／2	④
●ワープロ				
STYLO GRAPH	フルスクリーンエディタ＋英文WP	50,000円	59／2	⑤
●業務				
DAC財務会計システム	マルチタスクを活かした財務会計	198,000円	59／3	⑥
●簡易言語				
NOA 110	WP，作表，グラフ機能を総合した日本語データベース	未定	59／5	⑦

④マイクロウェア・ジャパン㈱　⑤㈱星光電子　⑥㈱第一情報処理センター　⑦㈱システムヤマト

2 1MBのミニフロッピ標準装備(×1)

FM-11 AD2にはBSに装備されている2HDのドライブが1台装備されている。これで，フロッピの容量は，320KB→1MBとなったわけで，ビジネスなどでも充分実用に耐えるものとなった。ただ，1台ではなかなか使いづらい点もあるため，できれば増設したほうがよいだろう。

3 F-BASIC V5.0 標準添付

FM-11 AD2には，これまでのF-BASIC V4.0の日本語処理機能を強化したF-BASIC V5.0が添付されている（もちろんV4.0も付いている）。

V5.0は次の特徴を持つ。

(1)漢字をBASICステートメント中の文字定数，REM文中，ファイル名に使える。
(2)カナ漢字，ローマ字変換が簡単に行える。
(3)ビジネス，漢字プリンタに出力できる。
(4) JIS, KNJ$, KLEN, KINSTR, KMID$, KACNV$, AKCNV$, KEXT$, KTYPE, KLEFT$, KRIGHT$など漢字用関数が追加された。
(5) SCREEN, CALL, SERCH, TALK文が新設（あるいは拡張）された。

CALLは，機械語プログラムへ複数個の引数を，スタック経由で渡せるコマンドで，

CALL　ADD (A%, B%, C%)

のように使える。

SERCHは，1次元配列中の指定データを検索するコマンドである。

TALKは，このたび新たに発表された音声合成ボードのサポートコマンドである。この音声合成は，大型コンピュータに使われているものと音質は変わらず，これまでよりもはるかに肉声に近いものとなっている。

4 日本語ワードプロセッサを添付

FM-11 AD2にはFM JWP／11　日本語ワードプロセッサがバンドルされている。

5 マウス(オプション)

AD2にも，マウスを接続できる（マウスに関してはBSの項参照)。

以上，FM-11 BS，AD2をざっと解説した。BS，AD2とも，これまでのFM-11のフロッピ容量を増し日本語機能を強化して，ビジネスユースに充分耐えるようにしたマシンであるといえよう。

なお，FM-11 BS，AD2の発表とともに，FMシリーズ用の周辺機器6種が発表された（**表3**)。

表3

製品名	型番	適用機種	価格	出荷
MIDIアダプタ	MB22439	FM-7	24,800円	2月下旬
音声合成カード	MB22441	FM-11BS, EX, AD, ST	29,800円	3月中旬
	MB22437	FM-7		
マウスセット	MB22440	FM-11BS, AD2, EX, AD, ST	34,800円	4月初旬
	MB22436	FM-7	33,000円	3月下旬
シリアルドットプリンタ	MB27409	FM-7, 8, 11	98,000円	2月下旬
カラーCRT(640×400)	MB27315	FM-11	118,000円	2月15日
ハードディスクユニット(20MB)	MB27653	FM-11	498,000円	2月15日

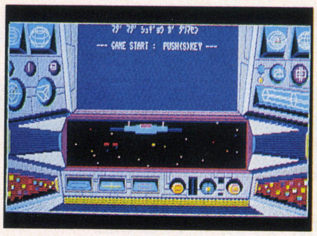
SOFT REVIEW

①ジャンル　②提供媒体（T：カセット，5D：5インチディスケット，8D：8インチディスケット）　③価格　④適用機種　⑤使用言語　⑥必要なオプション　⑦発売元

スターアーサー伝説II　暗黒星雲

①アドベンチャゲーム
②T（3本組），5D
③T：4,800円　5D：6,800円
④FM-7　⑤BASIC＋マシン語
⑥なし　⑦T&Eソフト

ついに第2弾が出た。レイソードを手にしたスターアーサーは，スーパーコンピュータ"テス"を備えたクランプトンIIを操り，レイソードの秘密を暴くべく，漆黒の宇宙空間へと旅立つ。今回はアドベンチャのほかに，3次元立体図形処理を用いたリアルタイム戦闘シーンも盛り込まれていて，このシーンをクリアするための"修業プログラム"まで入っている。グラフィックはきれいだが機内から見る宇宙空間の画面が小さく，戦闘シーンは，やや迫力に欠ける。今回は全体的に，宇宙シミュレーション的に仕上がっている感じだ。では健闘を祈る。



忍者秘剣伝　からくり屋敷の怪

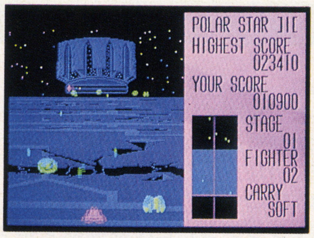
①ロールプレイングゲーム
②T　③3,500円　④FM-7
⑤BASIC　⑥なし　⑦エム・アイ・エー

秘剣「むらくも」のありかが記されている巻物を奪うため，不気味な屋敷に忍び込んだ君の前に，次々と現れる恐ろしい忍犬，毒ヘビ，侍……。君は耐久力，攻撃力，精神力のすべてで敵を圧倒し，巻物を探し出さねばならない。

ポーラースターIII

①アーケードゲーム　②T
③3,800円　④FM-7
⑤マシン語　⑥なし
⑦キャリーラボ

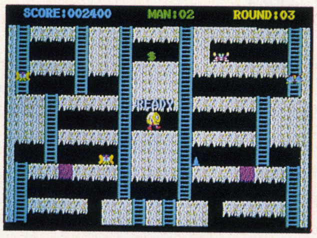
オールグラフィックによるスクロールゲーム。襲ってくる敵を攻撃しながら，最終的には敵の飛行要塞を破壊する。敵の飛来のしかた，スクロール処理などよく考えられており，とくに敵要塞が見えてくる場面などは圧巻。1面クリアするごとに敵までの距離がのび，難しくなっていく。迫力，面白さという点では太鼓判をおす。



Mr. HEADING(ミスターヘッディング)

①アーケードゲーム　②T，5D
③T：3,800円　5D：4,800円
④FM-7　⑤マシン語
⑥なし　⑦アポロテクニカ

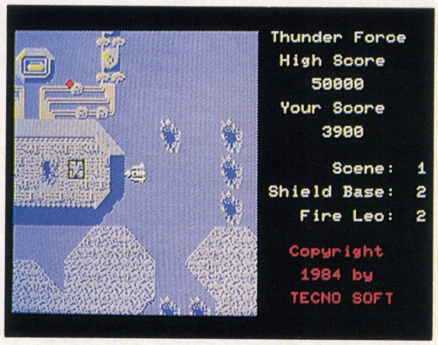
これもスクロールゲーム。ヘルメットをかぶったヘルメット君を操作して，地中を動き回るエイリアンの下へまわりこみ，ヘッドバット（頭突き）で攻撃。パターンは16面あり，各面ごとに難しくなる。何面かクリアするごとにコーヒーブレイクで可愛い女の子が登場。スペースフラッグなどの隠しコマンドも用意してある。よく考えて動かないと，エイリアンのはさみうちに合う。やや単調な感はまぬがれないが，楽しめる。



サンダーフォース

①アーケードゲーム　②T
③4,800円　④FM-7
⑤マシン語　⑥なし
⑦テクノソフト

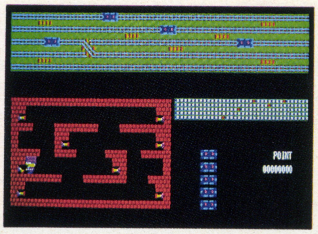
X1で大好評だったこのゲーム，FMへの移植版。4方向スクロールはX1と比べるとちょっと物足りない気もするけれど，FMの内部構造を考えたら，かなりの力作といえる。おそらくこれ以上スクロールを複雑にしたら，スピードの点で難がでるだろう。アーケードゲームはスピードが命だもんね。だからこのゲームは，FMのスクロールゲームとしては現時点では最高のものだ。技術的なことはもちろん，ゲームとしてもたいへん面白い。



パニック the トレイン

①アーケードゲーム　②T
③3,500円　④FM-7,8（共用）
⑤マシン語　⑥なし
⑦バックスエレクトロニカジャパン

次々に操車場に入ってくる列車を，衝突・脱線させないようにポイントを切り替える，というゲーム。ちょっと複雑なキー操作も覚えてしまえば簡単。手と頭を使わなければすぐ脱線。単純なゲームなんだけど，面白いんだよね，これが。操車室の中をウロウロしてると，本当にパニック状態になってしまう。まさにタイトルどおり。でも1つ注文。グラフィックはもう少し考えてほしい。アイデアはいいのにうまく生かしきれてないみたい。



NUTS＆MILK

①アーケードゲーム（ゲームプロセッサ付）
②T　③3,200円　④FM-7
⑤マシン語　⑥なし　⑦ハドソンソフト

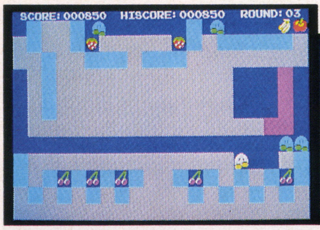
最近自分でパターンを作れるゲームプロセッサというのが増えてるけど，これもその1つ。MILKちゃんが迷路の壁を掘って大好きなフルーツを食べるんだ。でもおっと注意。NUTSくんが追いかけてくるし，壁は襲ってくるし，もうたいへん。こうした迷路のパターン，フルーツの置き場所，NUTSくんの数など好きなものが作れる。作ったパターン

はSAVEもLOADもOK。自分だけのNUTS & MILKができます。また，自分の好きな面を遊ぶこともできる。ゲームフリークにはちょっと物足りないかもしれないけど，入門者にはピッタリのゲーム……かな?

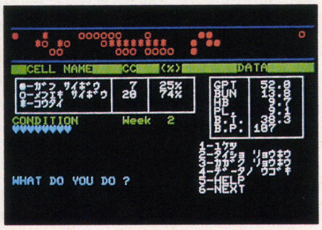
CANCER FIGHTING

①シミュレーションゲーム
②T ③5,800円
④FM-7,8 ⑤BASIC
⑥なし ⑦ダイヤモンド社

白血病患者をさまざまな薬を駆使して治療し，退院させる，というもの。ゲームといえども真面目にならなきゃ。マニュアルには薬の効果，各データの危険値など詳細に書かれており，誰にでもわかるようになっています。シミュレーションだから患者が病院から脱走したり，自殺してしまったりの事態も起きます。患者さんを助けてあげてください。しかし，そのうち人工受精ゲーム，なんてのができたら面白いと思いません?



エディタモードで
パターンをつくる⬇



ドイツ参謀本部

①シミュレーションウォーゲーム
②T ③4,500円 ④FM-7
⑤BASIC+マシン語 ⑥なし
⑦東映ビデオ

PLANNER 8

①簡易言語
②5D，8D（マニュアル付）
③155,000円 ④FM-11
⑤BASIC+マシン語
⑥漢字ROM，漢字プリンタ
⑦オービックビジネスコンサルタント

パソコンの質問に日本語で答えていけば，パソコンが目的のプログラムをBASICで自動的に作り上げるジェネレータ型の超簡易言語。経理，財務，総務，販売，購買，生産管理などのビジネスアプリケーションとして100件の事例が公開されており，あらゆる要求にきめ細かく対応している。

THE PG

①簡易言語
②8D（マニュアル付）
③98,000円
④FM-11
⑤BASIC
⑥なし
⑦ビジネスソフト

Just-PGの発展版。作成したプログラムは自分の開発したBASICプログラムとしてリストがとれ，追加・変更は自由に行える。4機能・8本のジェネレータより構成されており，目的別のプログラムが作れる。また，THE PGの機能拡張用として「THE PGプラス」（5D：4万円）が発売されており，複数ファイルのマッチング，集計，DMラベル作成処理も可能。

TRANS LETTER PROCESSOR

①英文ワープロ
②5D（マニュアル付）
③14,500円
④FM-7，8
⑤BASIC+マシン語
⑥フロッピ2ドライブ，プリンタ(全機種対応)
⑦マイクロテクノロジー

翻訳機能付英文手紙作成プロセッサ。手紙の種類を入力すると，書き出しと結びの定型文を好みの型に応じて自動作成する（18種の手紙のおのおのについて5つの文例)。通常の手紙文，とくにビジネスレターに多く使われる単語辞書は，約2,500語を収録。また，日本語のキーワードをカナ入力すると，自動的に英文に翻訳する機能もある（約4,500用例)。

漢字版 ザ・パソコン給与II

①給与計算
②5D，8D（マニュアル付）
③給与・賞与編148,000円
年末調整98,000円
④FM-11EX，BS
⑤BASIC+CP/M-86
⑥漢字プリンタ，EXは増設フロッピおよび漢字ROM
⑦ピーシーエー

給与，賞与，年末調整，人事管理を含む人事給与の総合システム。会社マスタ，個人マスタなどを登録すると月々，勤怠データなどの変更データを入力すれば，あとはパソコンが処理する。処理人数は5Dは300人，8Dは500人。対象者は日給・月給・時給・日給月給。部門は20。残業計算4種，所得税3種，支給項目16，控除項目14。総支給額は1人千万円未満。銀行振込，端数処理，賞与試算など実務機能が豊富。

FM-KOKYAKU

①顧客管理
②5D（マニュアル付）
③30,000円
④FM-7，8
⑤BASIC+アセンブラ
⑥フロッピ2ドライブ，漢字ROM，シリアルドットプリンタ，熱転写プリンタ
⑦エス・ビー・シー

登録件数は1枚のディスケットに3,840件までで，最大項目数は32項目。すべての項目に検索条件が設定できるほか，多重検索も可能。タックシールへのあて名書印刷はもちろん，一覧表は任意の項目を任意の順で自由に印刷できる。漢字入力は，カナ・ローマ字変換によるため簡単。漢字は6,000読み，熟語は14,000読みを収録。あらゆる業種に幅広く対応できるよう，データ入力画面，ファイルフォーマットが任意に設定できる。

Liaison（リエゾン）

①ローカルエリアネットワーク
②8D
③8台までの場合200,000円
④UCSD p-Systemが稼動するマシン
⑤UCSD Pascal
⑥(オムニネット，イーサネットなどの)ネットワークハードウェア一式，ミニ・マイクロコンピュータ（UCSD p-Systemの使用できるもの)，ネットワーク接続用のインタフェース
⑦日本ビジネスオートメーション

システムソフト（Liaison，ディスクサーバ，プリンタサーバ，ツールキット，移植キット）とアプリケーションソフト（データベース，電子メール，ワープロなど）の2つのソフトウェアに分けられる。異機種，異CPUでも同一ネットワークに接続すれば，プログラム，データ，周辺機器の共有化が実現。また，どのようなネットワークにも簡単に移植できる。

中高用時間割編成

①ビジネス
②5D（マニュアル付）
③80,000円
④FM-7，8，11（共用）
⑤BASIC
⑥フロッピ2ドライブ，シリアルドットプリンタ
⑦内田洋行

教員数は70名まで，クラス数は特別教室も含めた40クラスまで，教科数は講座授業も含む60教科まで。出力帳票は教師別時間割表，クラス別時間割表など。

成績管理

①成績処理
②5D（マークカード，マニュアル付）
③9教科用：60,000円 5教科用，15教科用：50,000円
④FM-7，11
⑤BASIC+マシン語
⑥フロッピ2ドライブ，シリアルドットプリンタ，マークカードリーダ(マミヤ製M300，SR301，M200)
⑦内田洋行

史上初の総力戦である第一次世界大戦をテーマとしたもので，あなたはドイツ軍参謀総長として同盟軍を指揮します。このテのゲームは状況判断，明晰な頭脳が必要とされるのですが，この「ドイツ軍参謀本部」では単なる状況判断で終わってしまっているのが残念。ただコンピュータが勝負を決めるのを待つだけという感じで，迫力に欠けます。かなり史実に基づいているゲームです。

ドニエプルリバーライン

①シミュレーションウォーゲーム
②T　③5,800円
④FM-7
⑤BASIC+マシン語
⑥なし　⑦東映ビデオ

1943年のソビエト軍によるドニエプル川侵攻を背景としたもの。あなたはドイツ軍司令官となって，コンピュータ（ソビエト軍）と戦うのです。ボードと駒が入っていて，ボード上で作戦をねらないとなかなかうまくいきません。がんばってください。

コーラボトラーズ

①ビジネスシミュレーションゲーム
②T　③4,500円
④FM-7,8
⑤BASIC+マシン語
⑥なし
⑦マイクロテクノロジー

コンピュータの経営するKING-COLA社に対抗して，あなたが社長としてFRESH-COLA社を1年間経営する。商品のシエルフライフの短いソフトドリンクビジネスの実際に即してシミュレーション。与えられた経営目標が達成されるまで何回もトライして，未来の大社長の第一歩を。

5教科，9教科用のものは，1クラス50名まで，1学年につき12クラスまでの合計600名までの処理。テスト回数は1教科につき25回まで合計225回（9教科用）。15教科用は1クラス50名まで，1学年20クラスまでの合計1000人。テストは1教科につき1回。集計方法は三者とも学級別，学年別，学籍番号順，成績順に作表し，成績表，度数分布表（5教科用はなし），個人表を出力する。

スポーツテスト

①成績処理
②5D（マークカード，マニュアル付）
③小学校用：50,000円
　中学校用：50,000円
④FM-7,11
⑤BASIC
⑥フロッピ2ドライブ，シリアルドットプリンタ，マークカードリーダ（マミヤ製M300，SR301，M200）
⑦内田洋行

1クラス50名まで，1学年12クラスまで。体力診断テスト1回，運動能力テスト2回。学級別，学年別，性別，種目別に集計作表（総合判定表，総合得点表，スポーツ記録表）する。マークカードには，体力診断，運動能力の2種類あり。

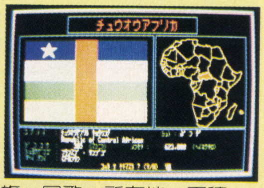
EXPERT MATHEMATICS 中学数学シリーズ

①CAI
②T
③各3,000円
④FM-7,8
⑤BASIC
⑥なし
⑦数研塾

中1用関数1（関数式とグラフ），関数2（比例・反比例），平面図形1（基本図形），中2用一次関数1（一次関数とグラフ），一次関数2（一次方程式と一次関数），図形1（角と平行線）。

たのしい世界の国々

①CAI
②T（2本組）
③4,800円
④FM-7
⑤BASIC
⑥なし
⑦T.S.Sスリー・スターズ・ソフト



世界の国々の国旗，国歌，所在地，面積，人口，主要言語，貨幣単位などその国の概要を手軽に楽しみながら学習できる。パソコンならではの音の出る百科事典といった仕組み。

メジャー（図形計測プログラム）

①科学技術
②5D（マニュアル付）
③25,000円
④FM-7,8,11
⑤BASIC
⑥デジタイザ（グラフテック製マイタブレットDT1000）
⑦グラフテック（旧渡辺測器）

図形をデジタイザで測り，位置，距離，方位，折れ線の長さ，直線の間の角度，多角形や円形の面積を計算する。画面とデジタイザのカーソルだけで操作できる。計算結果のプリントアウトも可能。

計測データ処理プログラム

①科学技術
②5D（マニュアル付）
③15,000円
④FM-7,8
⑤BASIC
⑥X-Yプロッタ（グラフテック製MP1000，FP5301など）
⑦グラフテック

多チャンネルのデジタル化したデータの原波形の再現および，チャンネル間の和，差，比の演算，演算処理結果のグラフ化などを行う。このほか各チャンネルまたは合成値の全体平均，区間平均，移動平均なども求めることができる。

室内レイアウトシステム

①CAD
②5D（マニュアル付）
③15,000円
④FM-7,8
⑤BASIC
⑥X-Yプロッタ（グラフテック製MP1000，FP5301など）
⑦グラフテック

キー操作により室内の家具，什器を自由に移動させ，配置を検討することができ，決定した配置に基づき平面図，透視図を作成する。

Insight Window Designer

①開発ツール
②8D
③50,000円（シングルコピー）
④UCSD p-Systemが稼動するマシン
⑤UCSD Pascal
⑥UCSD p-System
⑦日本ビジネスオートメーション

UCSD p-System上で使用するウィンドウ処理ライブラリ。複数のファイルをディスプレイに表示でき，ファイルからファイルへのデータ移動や照合ができる。表示は現在のレベル，メニュ1行，エラーメッセージの3行から成り，特別なハードウエアを必要とせず，カーソル移動キーによりメニュを選択する。文字表示によるのでビットマップ方式に比べ少ないメモリで，高速のウインドウ処理が可能となる。

PRINTER BUFFER

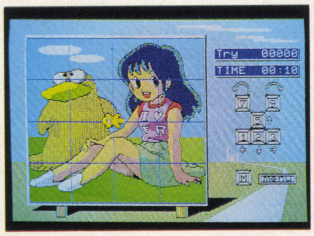
①開発ツール
②T（マニュアル付）
③4,800円
④FM-7
⑤マシン語
⑥なし
⑦エム・アイ・エー

これまでハードウエアで行っていたプリンタバッファ機能をソフトウエアで実現。割り込み処理でプリンタへの出力と同時に，プログラムのエディットや実行などの作業が可能になる。またBASIC命令で手軽に使用でき，プリンタへの出力命令（LLIST，HARDC，LPRINTなど）はすべてサポートしている。

TAAKOちゃん

①パズルゲーム ②T
③2,800円 ④FM-7
⑤BASIC+マシン語 ⑥なし
⑦ラポート

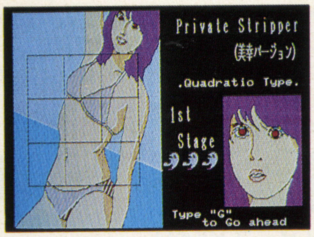
アニメファンなら知ってるよね，タッコちゃん。バラバラになったタッコちゃんとカモノハシの絵を，パズルみたいに動かしてきちんとした絵にするゲーム。絵の動き方が複雑で，ちょっとやそっとじゃ解けない。でも最後にはタッコちゃんが現れてくれるからご心配なく。どちらかといえばロリコンを自称する人に。根気よく遊んでネ。



女子大生プライベート

①パズルゲーム ②T，5D
③T：4,200円 5D：6,200円
④FM-7 ⑤BASIC+マシン語
⑥なし ⑦日本ファルコム

前出の「TAAKOちゃん」がロリコン向けなら，こちら「女子大生プライベート」はアダルト向け。美幸，優子，今日子の3人の美女の姿はなかなかリアルで，1人につき3ステージ用意されています。1ステージごとに絵はだんだん大胆になっていくのです。パズルはQuadratic TYPEとCubic TYPEの2つ。1度やるとすぐ飽きてしまうという難点はあるけれど，こういうのは思考過程が楽しいんだよね。あなたも女子大生のプライベートタイムをのぞいてみないか。



ガキ大将野球拳

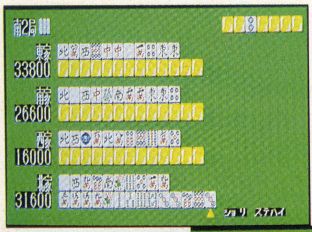
①思考型ゲーム ②T
③1,980円 ④FM-7，8
⑤BASIC ⑥なし
⑦セントラル教育

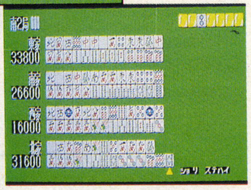
ロリコンブームのこの世の中に少々食傷気味のあなた，硬派と自称するあなた。強くてたくましい"男"と勝負してこそ本当の野球拳です。

ジャン狂

①ホビー ②T，5D
③T：4,000円 5D：6,800円
④FM-7 ⑤マシン語
⑥なし ⑦ハドソンソフト

本格的なマージャンゲーム登場。ルールはクイタンもしくは完全先づけを選択。東南まわしの一翻縛りで，5本場からはリャンシバ（二翻縛り）になる。オープンモードがあるので，初心者はコンピュータの手作りを見ながら対局することにより，手作りの参考になる。また点数計算は正確なので，点数計算の勉強にも最適。画面はきれいで見やすい。待たされたりせかされたりすることはないので，自分のペースで勝負できる。





オープンモードでコンピュータの手を見る➡

BIG PHONE
（連勝式勝馬予想プログラム）

①ホビー ②T
③5,800円 ④FM-7，8
⑤BASIC
⑥（コピーを取る場合）プリンタ
⑦ダイヤモンド社

競馬専門誌3誌のトラックマン，評論家，本紙予想5人ずつ計15人の評価を5段階の数字に変換し，各馬の持ち点として入力する。すると「BIG PHONE」が各枠ごとにA〜Hの評価をつけ，荒れる確率などによりおすすめの買い方を提示する。過去3年間の平均勝率は7割。

栄養カロリー計算と食事指導システム

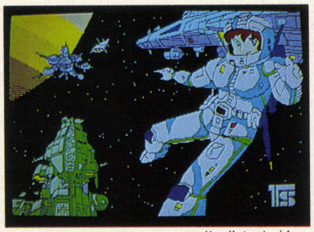
①実用
②5D（マニュアル付）
③9,800円
④FM-7，8，11
⑤BASIC+マシン語 ⑥なし
⑦マイクロテクノロジー

実際のメニューを入力するとそのカロリー計算と栄養分析をして，それぞれの過不足を指摘する。また食べたい食品を入力すると三食の適正配分や適正量を食品別に修正指示するとともに，成分分析を行う。

グラフィックツールセット

①グラフィックツール
②T（2本組・マニュアル付）
③4,500円
④FM-7，8（共用）
⑤BASIC+マシン語
⑥なし ⑦テクノソフト

ラインエディタとパターンエディタの2本より構成。自分で作ったBASICプログラム中に絵を書くところがあれば，そのプログラムをラインエディタのデータとしてLOADし，絵を書く部分だけを自動的に抜き出してくれる。しかもそれを改めてエディットすることも可能。また8原色を同時に異なった中間色に交換でき，自分がよく使う中間色を128個定義しておけば簡単に中間色が使える。方眼表示など使い勝手のよいグラフィックツール。



グラフィックツールで作成した絵

読者が選ぶゲームソフトベスト10（FM-7）

1位	スターアーサー伝説Ｉ 惑星メフィウス（T&E）
2位	ドアドア（エニックス）
3位	ジェルタ（キャリーラボ）
4位	ウルトラ四人麻雀（九十九電機）
5位	3Dゴルフシミュレーション（T&E）
6位	パックマン（電波新聞社）
7位	倉庫番（シンキングラビット） フロントライン（ニデコ）
9位	機動戦士ガンダムＩ ガンダム大地に立つ（ラポート） 信長の野望（光栄マイコン）

読者が選ぶビジネスソフトベスト5（FM-7）

1位	PN101（ワープロ・エスピーシー）
2位	JWP/7（ワープロ・エスピーシー）
3位	F漢（ワープロ・ニチコン設計）
4位	COMAS WP V2（ワープロ・コマス） 文太（ワープロ・CSK） 簡文（ワープロ・ADAオフィス）

▶アドベンチャゲームというのは，日本人にとってわかりにくいものだと思わされてきたが，「惑星メフィウス」の登場によって，そうした概念が破られた，といってもいいだろう。題材の面白さ，ストーリー展開，グラフィックや効果音などの重要性が充分に考えられている。時間と人手をかければいいものができる，という見本。「惑星メフィウス」以後，グラフィックに凝ったものが多くなったようだ。「倉庫番」もいいソフト。ゲーム自体はとても単純だが面白い。複雑なゲームの多い中で，シンプルさがうけた。思考型ゲームというパソコンならではのジャンルだったことも見逃せない。それまでのパソコンゲームのイメージを一新したと言える。

▶ビジネスソフトの上位は，奇しくもワープロでうめられた。一口にビジネスソフトといっても，さまざまなものがあるが，ワープロはその必要性，実用性からユーザにとっても受け入れやすいものの1つだろう。その傾向が今回の集計に現れていると思われる。1位の「PN101」は，昨年の11月に発売されたもので，18,000円と手頃な値段が魅力。2位「JWP/7」はFMにとって記念碑的ソフト。3位「F漢」はパッケージの面白さがうけた商品。4位「COMAS WP V2」は1月に発売されたものだが，75,000円とやや高め。「文太」「簡文」はテープベースのワープロ。ところでOh!FM 5月号ではワープロ特集を企画，現在作業進行中。乞うご期待。

特集　虫取り専科

バグ退治講座——宿命のライバル

川村　誠一

バクとはアルファベットでB,U,Gの３文字。そしてその意味は,虫のこと。けれどもコンピュータフリークにとっては，恐怖のキーワード。こいつときたら精神衛生上，百害あって一利なし，といえるほど憎たらしいヤツ。しかも神出鬼没だから始末におえない。この害虫駆除をデバッグと称し，駆虫道具をデバッガと呼んでいる。我々コンピュータを友とする者たちにとっては宿命のライバルといえるBUGとの対決は，さてどうしたものであろうか…。

第1章　敵の性質を知るべし

BUG，バグと大さわぎしても，相手のことをよく知らなければどうにもなるまいて，そこんところに注目してみよう。まず性格であるが，非常に冷酷であるものの，その行動パターンは一定である。だが，種類は多く，すべてのパターンを把握するのは並大抵なことではないのでR。しかして，虫取りの専門家と豪語する人の頭の中には，すべて入り得るそうな。そしてその行動パターンが分析できれば，先回りして防衛線を張ることができるのだ！もちろん例外もある。一定のパターンと思いきや，時にフライング的行動を示し，頑固に我々の目からのがれようとするヤツだ！こいつは本当に曲者。でもご安心めされ。何といっても相手は機械で，こちらは知恵持つ人間ゾ，負けるわけがないのだ。ただし感情がある点，気分によって左右される……etc.は，人間様の欠点であろう。そしてその一瞬を盗んで虫は発生してしまうのだ。何も機械が悪いわけではないのだ。ホント。

つい感情に走ってしまったが，今一度，ヤツラの性質を確認してみよう。

❶**冷酷である**（発生する条件さえ揃えばどんどん無慈悲に発生する）

❷**パターンは一定**（この虫は夜行性，あの虫は肉食などと種類によって，ほぼ行動パターンが決まっている。逆に行動パターン，すなわちエラー内容から名前まで言い当てることが可能）

❸**例外もある**（いちばんやっかいな虫。たいていは人間様にミスがある。基本的な考え方の誤り，アルゴリズムの間違いなど，ひどい場合は他人によってのみ発見されるゲーム虫などの重症なものまである）

❹**見えない敵となることがある**（これもやっかいな虫の１種。プログラムに構造的な誤りはないし，エラーが出て止まってしまう，なんてことも生じない。だが，ひとたび製作者より離れて他の人に渡ったときに出てきやすい虫である。いわゆるリストの打ちマチガイ，入力ミスによるものでcarelessnesとも言われ，全面的に人間様の誤り。未然に防ぎ得る，もっとも簡単なものである）

大別すれば上記の４つに分けられると思う。これらを念頭において，虫取りに精だしてみよう。

第2章　まずはエラー内容を確認せよ！

BEEP音とともにプログラムはSTOP。○○× Error in □□□の表示。「○○×？　ははあーん，あそこをミスったかな？」などと大まかな予想を立てておもむろにLIST □□□◯。「あっ，やっぱりネ。」

とまあ一般的には，これら一連の作業を繰り返すわけだが，この作業が虫取りの基本中の基本といってもいい（誰でも知っていることだと思うけど）。改めてエラー退治の手順を考えてみよう。

その１　エラーメッセージを出力しよう（虫さんはドコ？　種類は何か？）

エラーが発生すると(BASICプログラムのみの場合に限る。マシン語のサブルーチンを使用しているBASICプログラムでは，必ずしもそうではない)，まずはBEEP音が聞こえるハズ。と同時に，エラーメッセージがCRTに表示されていて，Readyとカーソル点滅がプログラムの中断，コマンド待ちの状態を示している。

これが当たり前だと思っているあなた！　すでに虫さんの思うツボなのだ。そう，必ずしもエラーメッセージが出ているとは限らないのである。そんな（エラーメッセージの出ない）プログラムは，プログラム中でCONSOLE文を使ってませんか？　たとえばOh！FM 4号「もぐらたたきゲーム」の80行。

```
80 WIDTH80,25:CONSOLE24,1,0,0:COLOR0,5:CLS
```

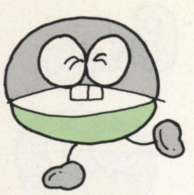
もちろんCONSOLE文のスクロール範囲にもよるけれど，少なくとも「もぐら……」ではBEEP音とカーソルの点滅だけしかわからない。そんなとき必要な命令はたった２つ。それは，

```
WIDTH 80      または      WIDTH 40

LIST.
```

もちろん省略型でも可能。すなわち，

```
W. 80      または      W. 40

L.
```

これでエラーのあった行が確認できる。ただしこれでは，エラーメッセージは表示してくれない。そう，どんな虫かわからなければ，デバッグのしようがない。虫さんはいつまでも元気！「最初のCONSOLE文など取ってしまえばいい。俺はいつもそうしてるぜ。虫が全部とれてからそこだけ加えるのサ！」



ごもっとも。確かにこの方法は簡単だけれど……「エラーの出てない行に手を加えるのは，別のエラーを発生させる原因になりかねないから，イジラナイほうがイイのだ」。これもまったくそのとおり。何も好きこのんで虫をばらまく危険な種を植える必要はない。ここで威力を発揮してくれるのが次に示す2つの予約語*1。すなわちERRとERL。これは正確には予約変数ともいうべきもので，その内容は変化する。

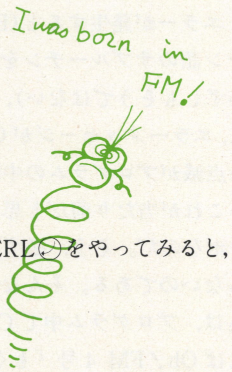
ERRはエラーコードを，ERLはエラーの行番号をそれぞれ値として持っているのだ。これは非常に便利な機能なんだが，知ってる人，使ってる人は案外少ないみたい。とくに初心者マークのついている人は，これを機に覚えてしまおう。プログラムがエラーでストップしたとき，

```
PRINT ERR,ERL
```

を実行してみよう。

下のリストを見てくれ。明らかに20行にエラーがあるよね？ Type Mismatchというやつである。実行しなくてもわかると思うけど，もし実行すれば次のようになるハズ。

```
10 INPUT IN$
20 A=IN$

Ready
RUN
? 1

Type Mismatch In 20
Ready
■
```



ここでさらに先ほどのPRINT ERR，ERLをやってみると，

```
PRINT ERR,ERL
 13             20

Ready
■
```

となる。ERRには13，ERLには20が入っていたわけ。エラーコード13をエラーリスト表で調べてみると，ありました。Type Mismatchと書いてある。そしてエラー行は20行であるわけね。それではプログラムを実行していないときのエラーの場合はどうか。65535を値として持ってしまうのでR。実際に以下の例をやってみてネ。

```
A
Syntax Error
Ready
PRINT ERR,ERL
 2              65535

Ready
■
```

ERR＝2はSyntax Errorのコードで，ERL＝65535はすなわちダイレクトモードであることにほかならない。

*1**予約語**：BASIC中でBASIC自身が定義し使用しているもの。たとえばLISTとかIF，PRINTとかいうやつ。これらは原則として変数名などには使用できない。

図1　エラー関係コマンド

```
TRON        TROFF
STOP
CONT ――――――――STOPで止まったプログラムを次の命令から実行
ON ERROR ――――エラーが起きたら分岐
PRINT ――――――変数名，パラメータの表示
```

以上，WIDTH，LIST，ERR，ERLの4つを使えば，エラーメッセージが確実に見られる。お試しあれ。

その2　エラー関係コマンド詳解

○ **TRON, TROFF**

これはBASICのトレーサとも言えるもので，マシン語のプログラムを1ステップ（1命令）ずつ実行して，そのつどレジスタの内容を確認するものと，とてもよく似ている。ただTRONの場合は今実行している（正確には次に実行する）行番号を表示し，その行を実際に実行していく。もちろん[ESC]を押して，ちょうど1行ずつ実行させることも可能。ところがいわゆるマシン語のトレーサと違うところは「行番号の表示」と「その行の実行結果」のみ，という点なのだ。マシン語のトレーサでは，ちょうど行番号にあたるプログラムカウンタやアキュムレータ，インデックスレジスタ，スタックレジスタ，コンディションコードレジスタすべてを表示してくれる（**リスト1**）。

「BASICでもそれくらいやってくれてもいいんじゃないか！」なんて怒らないでネ。というのもマシン語プログラムのときにCPU内で扱える変数名は，各レジスタの数だけ。CPU内のレジスタ数は，6809ではA，B（A＋BでD），X，Y，U，S，CC，PC，DPの9個なので，それらすべてを表示するのはいとも簡単。また逆にCPU内での動きを見るために絶対に必要なことなのでR*2。その点でBASICではたくさんの変数を定義使用できるため，すべてを表示するのはまったく不可能，とは言わないけれども，たいへんなことなんだ。また，そんな必要もない。というのも，BASICなら変数の内容を見ることは簡単だろう？　それよりもちゃんと実行しているか，思ったとおり分岐しているだろうか，っていうほうが大切になってくる。そう，行番号とその行の実行だけでも，虫はたくさん取れてしまうんだ。だからどんどん使ってみることを勧めるナ。えっ，使い方がわからないって？

それならば，有効な使い方を教えちゃおう！

2つの使い方があるけれど，まず基本的なものから。

❶ダイレクトモードで使う

という奴だ。そう，コンピュータが，

```
Ready
■
```

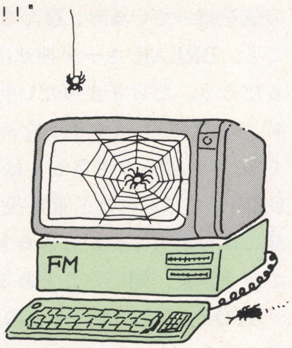
*2ここで誤解のないようにしてほしいのは，実際にはCPUには広大なメモリエリアが存在しているわけで，変数として扱える領域はメモリエリアの大きさに依存している。が，メモリエリアの変数エリア（変数として扱い，かつ格納してある領域）を決めるのは人間。だからCPUが知っている変数名，すなわち扱える変数名はCPU内のレジスタの個数だけなのネ。

とCRTに表示しているときがそうだよネ。プログラムを実行していない，例のアノときですよ。このときにおもむろにTRON⏎と入力してやると，アレレッ，またReadyが出て何もしてくれなかったみたいになってしまった……。

```
Ready
TRON

Ready
■
```

でもご安心を。コンピュータの方ではしっかりと覚えてくれているので，この状態からさらに，RUN⏎と入力してプログラムを実行させれば，あ～ら不思議。行番号の表示と実行をちゃんと行ってくれるじゃないか！

```
LIST

10 PRINT "TRON"
20 PRINT "TEST TEST TEST!!"
30 GOTO 10

Ready
RUN
[10]TRON
[20]TEST TEST TEST!!
[30][10]TRON
[20]TEST TEST TEST!!
[30][10]TRON
[20]TEST TEST TEST!!

Break In 20
Ready
■
```



先ほど言ったとおり，コンピュータはTRONを実際に実行してくれたわけなんだ。考えてみたら当り前だけど，これを理解するのって並大抵のことじゃないんだ。もう一度言うとTRONとは「BASICプログラムをトレースする命令で，実行する行番号を表示してそのあとにその行を実行する」こと。

さて，また別の使い方として，

❷プログラムの中で使う

これすなわち間接モードで使うということ。例を挙げよう。

```
5 TRON
10 PRINT "TRACE ON MODE NOW !!"
20 FOR I=0 TO 100 : NEXT
30 PRINT "CONTINUING.."
40 FOR I=0 TO 100 : NEXT
50 BEEP
60 GOTO 10
```

これを実行した結果が下の例である。

```
Ready
RUN
[10]TRACE ON MODE NOW !!
[20][30]CONTINUING..
[40][50][60][10]TRACE ON MODE NOW !!
[20][30]CONTINUING..
[40][50][60][10]TRACE ON MODE NOW !!

Break In 10
Ready
■
```

TRONが5行に入っているので，5行は表示されていない点に注目していただきたい。さっき言ったように，TRONは行番号を表示してからその行を実行するので，5行は表示されないのでR。もちろん60行のGOTO10をGOTO 5に変えれば，RUNさせた最初以外は5行も表示されることになるわけだ。確かめてみてネ。

さて上のプログラムに，

```
55 TROFF
```

を挿入したらどうなるであろうか。もうここまでくればおわかりだと思う。すなわち，60行は表示されず，それ以外の行（55行が追加される）が表示されたあと行番号を表示しなくなる。TROFFはその命令が実行された直後から，TRONの命令を解除してくれる。下がその実行例。

```
RUN
[5][10]TRACE ON MODE NOW !!
[20][30]CONTINUING..
[40][50][55]TRACE ON MODE NOW !!
CONTINUING..
TRACE ON MODE NOW !!
```

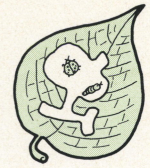
リスト1

```
**SET P=3000
**R
CC=00 A=00 B=00 DP=00 X=0000 Y=0000 S=C07F U=0000 N=00 P=3000 PSHS #$06
**TSIM 6
CC=00 A=00 B=00 DP=00 X=0000 Y=0000 S=C07D U=0000 N=00 P=3002 LDA  #$10
CC=00 A=10 B=00 DP=00 X=0000 Y=0000 S=C07D U=0000 N=00 P=3004 LDB  ,X
CC=08 A=10 B=FF DP=00 X=0000 Y=0000 S=C07D U=0000 N=00 P=3006 MUL
CC=09 A=0F B=F0 DP=00 X=0000 Y=0000 S=C07D U=0000 N=00 P=3007 STD  $300C
CC=01 A=0F B=F0 DP=00 X=0000 Y=0000 S=C07D U=0000 N=00 P=300A PULS #$86
CC=01 A=00 B=00 DP=00 X=0000 Y=0000 S=C081 U=0000 N=00 P=9464 NEG  $00
**■
```

"TROFF"について

スイッチにオン，オフがあるように，TRONにもそれを止める命令がある。それがTROFF命令である。TROFFもTRON同様に，その命令を一度実行してしまえばずっとそのモードになる。このTROFFは，TRONとペアなので，ここで覚えてしまおう（ちなみに，MOTOR命令は，それだけだと1回命令するごとにモードが変わる。このようなものをトグル動作という）。

Syntax



```
CONTINUING..

Break In 30
Ready
■
```

ここで挙げたプログラム例は説明のため簡単なものになっている。だけど，実際にTRONを使ったことのある人なら知っているよネ。CRTの表示を目で追うなんて，無理な話。プログラム中のLOCATE命令やCLS，WIDTHなどでCRT上はメチャメチャ！　行番号がぐちゃぐちゃとCRTを埋めてしまったりする。行番号を出力する際，BASICでは自分の中のプリントルーチンを使っているため，プログラム中のLOCATE命令やCLS，COLORなどCRT上に何かする命令にすぐ影響を受けてしまうのである。「プリンタに打ち出せたらナー」と思っている人は少なかやずいるハズ。そんな人に朗報がある。次のLTRON（エルトロン）がそれ。

○ LTRON

念のため言っておくけど，F-BASICにLTRONなんて命令はない。これは僕が考えたもので，プログラムにたった一行，文字数にしてたった11文字追加するだけでLTRONが実現する，そういうことなのですね。で，手順はというと，$5AC番地に0以外を書き込むことなのだ。つまり，

```
OO POKE &H5AC,1
```

(OO：プログラムの先頭行番号)

とすればよいのだよ，諸君！　ただしFM-7，8での話。11の人は我慢していただきたい（7，8はソフト的にもハード的にもほとんどコンパチブルなので，内容の解析はかなり進んでいる。が，11はいまいちというところなので，今回は見送らせてもらう）。この1行をプログラムの先頭に加えるだけでいいのだ。とても簡単。ウソだと思ったらやってみて。これでCRT上のゴチャゴチャTRONはキレイに片付いてしまうのだ。これは便利ですよ！

○ STOP

TRONもLTRONも，どちらも行を追ってくれるわけだけど，「○×行は本当に思いどおり動いたのだろうか？」なんて疑問に思ってる間もなく，プログラムはどんどん先に。STOP命令は，そんなときに有効的。STOPとは文字通り“一時停止”のこと。ほら，道路標識にあるアレョ（アメリカ映画なんかによく出てくるヤツ）。

これをプログラム中に置いてやる。するとプログラムはBREAK(またはSTOP)キーを押してもないのに，あーら不思議。止まってしまうのだ。TROFF ⏎と入力して，先ほどのプログラムのそうだな，54行あたりにSTOPと入力してみよう。RUNさせると下のようになる。

```
LIST

5 TRON
10 PRINT "TRACE ON MODE NOW !!"
20 FOR I=0 TO 100 : NEXT
30 PRINT "CONTINUING.."
40 FOR I=0 TO 100 : NEXT
50 BEEP
54 STOP
55 TROFF
60 GOTO 10

Ready
RUN
[10]TRACE ON MODE NOW !!
[20][30]CONTINUING..
[40][50][54]

Break In 54
Ready
■
```

ほら止まっちゃっただろっ！　キーには指一本触れないでプログラムが止まってしまうのである。これを自分が確かめてみたい行の次あたりに入力しておけば，自動的にプログラムは停止するので，たとえばある変数がどうなっているか，自分の思いどおりの値を持っているか，なんてすぐに調べられる。「なーんだそんなこと，BREAKキーを押せば済むことじゃないか」という人もいるだろう。だけど止めたい行で止めることができるかナ？　ダメダメ，INPUT文や小さなルーチン内なら止められるなんて言っても，それはそうやさしくはないからネ。このSTOPってやつは，BASICではそれほど重要視されていないみたいだけど，機械語開発には必要欠くべからざるものなんだ。「ブレークポイント*3」っていう言葉，聞いたことあるかな？　これすなわち，機械語中の「STOP」設定みたいなもの。でも「ブレークポイント」という名前の命令があるわけではなく，あくまでもある点（メモリのあるアドレス）でプログラムの実行を終えたがごとく振る舞ってもらう疑似処理のことなんだ。その詳しい内容は別の機会にするとして，BASICのSTOP命令を見直してみていろいろ使ってみるとよいと思う。**リスト2**を見てほしい。

これは二次方程式の根を求めるプログラム。たとえば，

$x^2+2x+1=0$

これは，

$(x+1)^2=0$

だから，

$x=-1$（重根）

になるはずなんだけれども，**リスト2**でやってみると，

$x=0, -2$

おかしい，実におかしい結果が出ている。ここで35行にSTOPを入れてみる。

するとA，B，CをINPUTした直後で終わるので，A，B，Cの各数値，FNAの内容がわかるはずだ。やってみよう。

*3ブレークポイント：機械語の場合は，BASICのようにエラーメッセージなどなく，あるのは悪名高き「暴走」のみ。その暴走を止められるのは唯一「リセット」だけなんだけど，リセットがハード的に強制的にCPUを止めてしまうのに対して，ブレークポイントというのはソフト的になんとかCPUに歯止めをしようとして設定してあげるもの。機械語のデバッガには必ずこの機能がついている。

```
A,B,C? 1,2,1
Break In 35
Ready
PRINT A,B,C,FNA(A,B,C)
 1              2
 1              4
```

A，B，Cの値は確かに正しい。けれども，あれ？ FNAの値がおかしい。この係数の二次方程式では判別式，すなわちFNAの値は0にならなければならない（重根だから)。判別式は，

$$D=B^2-4AC$$

だからFNAをみてみると……，あっ，違っている。数学のときのクセのまま，

FNA (A, B, C)＝B＾2－4＊AC

と書いてしまっている。ACでは「AかけるC」を意味せず，「AC」という変数になってしまっている。これでは何のために引数を3つも用意したんだろう。B^2の結果しか出てないではないか！で，直したものが**リスト3**。

これでやっとちゃんとした答え（思ったとおりの答え）が得られたわけだ。とまあ，こんな具合に使える。もう一度言うと，STOP命令で止めて，その直前の変数やパラメータなどをCRT上で確認する。

次の虫取りには，ぜひ使ってもらいたい方法の1つだネ！

第3章 よくでる虫の駆除法

これが本命中の本命であり，かつ虫取り術の基本のまた基本であるので，しっかり把握してもらいたいところ。一口にエラーといっても種々雑多。ここでは誰でも体験したであろうSyntax Error, Illegal Function Call, Subscript Out Of Range, Data-Readエラーの4つに焦点をあててみた。目を皿にして読んでほしい。

1. 何といってもSyntax虫

これはもう，超ド級宇宙戦艦，横綱級のものでありましょう。これを知らずして，何がエラーか！ ゴキブリと同じくらいにマイコン庶民には知れわたり，親しまれ，憎まれているといっても過言ではありますまい（なのにエラーコードでは2番目に番付けされている。どうも合点がいかないと思っているのは，僕だけであろうか？)。この虫，少しプログラムに慣れている人にとってには，逆に安堵ともいえるエラーメッセージだけど，初めての人にはさっぱりわけのわからない，正体不明のヤッカイな虫。たとえば，「A＝1」。これは「変数Aにプラス1を代入する」という意味で，正しいんだけど，「1＝A」……こいつはいけない。めでたくSyntax虫の発生と相なるわけ。この何の変哲もない，「数学的に見れば正しい」と思ってしまうところに，虫は出てきてしまうのだ。

ね，BASICを理解しているなと思った人にはわかるでしょ？わからない人のために説明。これは「Aに1を入れる」という代入式なのです。本来は「LET　A＝1」が正解で，大昔のBASIC(知っている人は通とみた！）では，省略不可だったのだ。

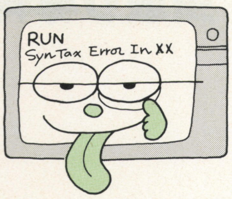
LETは知ってるよネ？ 中1の英語だョ。「～させる」という意味のアレさ。だから，昔BASICをやった人の方が早く納得できたし，間違えにくいようだ（やっぱり，イコールだけだと数学的発想になっちゃうんだよね，これ)。でも，最近のパソコンのBASICは省略可能になってしまった。一度わかってしまえば，LETなんて3文字は冗長すぎるし，面倒。それでも！ こういう，一般大衆にパソコンが浸透した今こそ，あってしかるべきだ！……と思ってしまう。まあここで，LETの是非について語っても仕方ないので打ち止め。

リスト2

```
10 CLS
20 DEF FNA(A,B,C)=B^2-4*AC
30 INPUT "A,B,C";A,B,C
40 IF FNA(A,B,C)<0 THEN 100
50 PRINT "KAI1=";(-B+SQR(FNA(A,B,C)))/(2*A)
60 PRINT "KAI2=";(-B-SQR(FNA(A,B,C)))/(2*A)
70 GOTO 30
100 PRINT "KAI1=";-B/(2*A);"+";SQR(-FNA(A,B,C))/(2*A);"i"
110 PRINT "KAI2=";-B/(2*A);"-";SQR(-FNA(A,B,C))/(2*A);"i"
120 GOTO 30
```

リスト3

```
10 CLS
20 DEF FNA(A,B,C)=B^2-4*A*C
30 INPUT "A,B,C";A,B,C
40 IF FNA(A,B,C)<0 THEN 100
50 PRINT "KAI1=";(-B+SQR(FNA(A,B,C)))/(2*A)
60 PRINT "KAI2=";(-B-SQR(FNA(A,B,C)))/(2*A)
70 GOTO 30
100 PRINT "KAI1=";-B/(2*A);"+";SQR(-FNA(A,B,C))/(2*A);"i"
110 PRINT "KAI2=";-B/(2*A);"-";SQR(-FNA(A,B,C))/(2*A);"i"
120 GOTO 30
```



話を戻して，この他にも Syntax 虫はちょっとしたことでも大量発生してしまう。以下に箇条書きしてみよう。

i)変数名の頭はアルファベットか？

ii)変数名中に F-BASIC のキーワードが入っていないか？

iii)ERR，ERL などの予約変数に値を代入しようとしてないか？

iv)行番号が許容範囲内か？

v)IF 文中で THEN がないのに ELSE を使用していないか？

vi)関数の引数の数は正しいか？

などである。次に，その説明を詳しくしてみよう。

まず，i)であるが，「変数名として許される文字は，英字および数字ですが，最初の文字は英字でなければなりません」なのであり，7，8では最初の16文字，11では40文字まで認識されるものでなければいけません！　それ以外は絶対ダメなのである。「そんな厳しいこと，言わんといてーな！」などとコンピュータに泣きついても，血のない相手には温情など無駄なの。だから，しかと頭に入れておくように。この例を挙げておこう。

```
10 ANTAGATADOKOSAHIGONANODA=6809
20 PRINT ANTAGATADOKOSAHIGODENAI

Ready
RUN
 6809

Ready
■
```

```
LIST

10 INPUT 3A

Ready
RUN
? 123

Syntax Error In 10
Ready
■
```

次にii)。F-BASIC のキーワードとは，F-BASIC 文法書中にあるすべての命令語のことで予約語というやつ。これは F-BASIC があらかじめ「私が仕事上使い，かつ理解している語彙は，すべて予約語または予約変数という。そしてユーザはそれらをユーザの変数名などに使用してはならない。なぜなら私は，自分の理解している語とユーザ定義の語との区別がつかなくなり，仕事ができなくなるからである」と宣言したものなので，賢物のコンピュータにこう言い切られてしまっては，我々人間サイドはそれに従うしかないのである。実際，コンピュータ側では予約語，予約変数名を自分のある領域上にもっていて，プログラムを実行するときに1つ1つそれらすべてを確認のうえ，おのおのの処理を行っている。

たとえばリストをとるとき，ちゃんと4文字 LIST または2文字 L. と入力したうえで↵を押さなければ，絶対に実行してくれないよね。前者はフルネーム，後者は省略形だけど，どちらも過不足があってはダメ！

“予約語”について

BASIC の予約語，予約変数はなぜユーザが使用してはいけないのだろうか？

まずは下のリストを見てほしい。

```
10 PRINT "PRINT ﾙｰﾁﾝﾃﾞｽ｡"
```

上のリストはメモリ上にたとえば次のように格納されている。

```
0790 07 A7 00 0A B9 20 22 50
0798 52 49 4E 54 20 D9 B0 C1
07A0 DD C3 DE BD A1 22 00 00
```

1．次の行のトップアドレス（2バイト）

2．行番号（2バイト）

3．行の内容

という順序で格納されている。上のダンプでは，$790，$791番地の07A7が1.に当たり，$792，$793番地の000A が2.すなわち行番号（10進数で10）に当たっている。それ以降（$794番地以降）が10行の内容になっているわけである。これを ASCII コードに直すと（上の□で囲まれた部分），

```
ｹ "PRINT ﾙｰﾁﾝﾃﾞｽ｡"
```

のようになる。BASIC のもとのプログラムと少し違う。PRINT 命令がなくなっているかわりに，片方のカナの「ケ」が出力されてしまっている。それ以外は全く同じである。キャラクタコード表を見ても，確かに $B9は「ケ」になっている。が，RUN させるとちゃんと10行を実行して，PRINT してくれる。この「ケ」が，いわゆる「中間コード」である。この「中間コード」というのは，BASIC が命令語をメモリ上におくとき，かつ実行するときに使われるもので，なぜこんなことをするかというと，

1．省メモリ化できる

2．実行時解釈の省略化

3．実行時スピードの高速化

が目的だからである。この「中間コード」を持っているものが「予約変数，予約語」なのである。再び下のリストを見てほしい。

```
10 PLINT "PRINT ﾙｰﾁﾝﾃﾞｽ｡"
```

Syntax Error になる。PRINT の「R」が「L」になっている。このプログラムは次のように格納されている。

```
0790 07 AB 00 0A 50 4C 49 4E
0798 54 20 22 50 52 49 4E 54
07A0 20 D9 B0 C1 DD C3 DE BD
07A8 A1 22 00 00 00 00 00 00
```

さきほどのように，□で囲まれた部分を調べてみると，

```
PLINT "PRINT ﾙｰﾁﾝﾃﾞｽ｡"
```

となる。この場合，ダブルクォーテーション内の文字のほかに，プログラムの命令（エラーであるが…）も，何もかも ASCII コードでしまってあるわけだ。すなわち，中間コード化されてないのです。

コンピュータ側では，ダブルクォーテーションで囲まれた文字（もちろんダブルクォーテーションも）と変数名以外は命令語として処理をして，1または2バイトの中間コードに変えてしまうのである。

つまり，中間コードに変えられるすべてを「予約語」，「予約変数」というわけである。

逆にいうと，上の PLINT のような中間コードを持たないものは，変数名として理解する。変数名の次にダブルクォーテーションが出てくることはないので，Syntax Error（文法上のエラー）となるわけだ。同じように，本文中にもあるが，SYNBOL を SYNBØL とミスタイプすると，中間コードに変換されず，変数とみなされ，（X，Y）の2次元配列変数となる。当然，型宣言文（DIM 文）がないので Subscript Out Of Range が出てくる。

```
LIST

Ready
L IST

Syntax Error
Ready
■
```

融通がきかないのはコンピュータの宿命*4（人間同士ならば，"リスト"でも"リースト"でも"トスリ"でもそれがLISTのことだなってわかるけど，現在僕らの目の前にあるパソコンでは無理なのである。でもこの融通のなさが逆に厳格さ，正確さ，偉さに通じているみたい）。すなわちコンピュータ側では"リスト"は"LIST"または"L."の2通りでしかなく，「これは私の知っている語彙の中にあるのだろうか」とすべて検索し，該当しているものがあるかどうか，シラミつぶしに捜してしまう。だからBASICはスピードが遅いのだし，逆に暴走しないわけ。

BASICが捜してもキーワードが見つからないようにするには，頭の1文字を変えてしまうのがてっとり早い。たとえばLISTを変数として使いたいならば，RISTに変えてしまうとか（日本語ではどちらもリスト。わかりやすいネ）。もちろん頭に1文字付加したりしてもいいわけだよネ（ただしLLIST(エルリスト)なんてのには注意！）。とにかくキーワードには常に注意をはらうようにすること。自分で作ったプログラム中で使う変数名などは一覧表にして，キーワードから始まっていないかチェック。キーボードからの入力のときにうっかり別のものにしてないか，などに気をつければ，キーワードの誤用虫は発生しないハズ。

iii）もii）と同様だけども，予約語ではなく予約変数という点が違う。予約変数には，TIME$のようにユーザーが値を代入できるものもあるが，ERRやERLのようにユーザーが値を代入できないものが多い。ERRやERLに値を代入してはいけない，これも絶対に守らなければならない約束の1つだネ。約束破れば，即，虫発生である。

iv）もちょっと注意。行番号は0～65535までの65536個使えるはずだけど，機種によって違うから。たとえばFM-7，8では0～63999の64000個，FM-11では0～65529の65530個，さらにほかのBASICでは別の値をとっていることがある。だからFM-11で64000行以降にサブルーチンか何かを置いといて，うまくいっているプログラムでも，FM-7や8では動かなくなってしまう。

```
LIST （FM-11の場合）

64000 '*** 5 KEY IN ROUTINE ***
64010 COUNT=0 : IN0$=""
64020 IN1$=INKEY$
64030  IF IN1$= THEN 64020
64040 IN0$=IN0$+IN1$
64050 COUNT=COUNT+1
64060  IF COUNT<5 THEN 64020
64070 PRINT IN0$

Ready
RUN
12345

Ready
■
```

（FM-7，8の場合）

```
AUTO 64000

Syntax Error
Ready
■
```

ね，FM-11では動いているけれど，7や8ではダメ。こんなときどうするか。そう，RENUM命令でもっと若い番号にして，その機種の範囲内に収めてあげればいい。そのほかにもマルチステートメントを多用して行数を減らすとかもあるけれど，それよりもプログラムを分割するとか，MERGE，CHAINなんてのを覚えた方が得策だろう。何？　全部の行を使って，しかもマルチステートメントをすべての行に使って，1文字も詰め込んだり修正できないって？　そんなときはNEWにしましょ！

v）のエラーはよほどでなければ起きないものだけれど，とくに多いのは先述のi）やii）の絡みだと思うな。まったく素直にIF文中でTHENを忘れるなんてことは，あまり起きないことだろう。けれども下の例みたいなのは，果たしてどうだろう？

```
10 INPUT A
20 IF A=0 GOTO 100 ELSE 200
30 END
100 PRINT "0 デスネ!" : END
200 PRINT "0 デナイ!" : END
```

「THENがないのにELSEがある，だからエラー」と答えた人いるかな？　実はこれエラーになりません。短いからぜひ試してみよう。20行のGOTOをTHENに変えてもらえばわかると思うけど，THENの方は分岐する行番号も許可しているし，実行する命令を置いてもいいわけなんだよね。さっきのように答えた人は，もう一度マニュアルのIF文のところを読み直してほしい。

話を元に戻して，**リスト4**を見てください。どうだろう，エラー出ると思いますか？　出ませんよ！　それでは**リスト5**ならどうだろう？　エラー出るかな？　これも正しく，エラーは出ません。ところが**リスト6**ではエラーになってしまう。20行でみごとに，Syntax Errorとなる。どこだかわかる？　そう，**リスト5**を**リスト6**に変えたらエラーが出たわけだから，**リスト6**で追加した5行か20行のどちらかにある。でも5行では起きなかったから，命令や変数は正しいハズだよネ。で，それならどうしてだろうと20行を見ても「BORDER」という変数を今までの20という定数に置きかえただけで，つづりに間違いはない。強いていうならTHENにスペースをあけずにくっつけてしまったことぐらいだ。実はこれがいけないのだ。賢明な読者ならお気付きだろう。そう，20行ではBORDERという変数は存在せず，BORDERTHENという変数が存在してしまっている。しかもそれだけではなくて，

*4この融通がなんとかきくようにしようというのが，最近巷をにぎわせている第5世代コンピュータ開発っていうやつ。

Ille 虫

リスト 4

```
10 INPUT ANTANONAMAE$
20  IF ANTANONAMAE$="MY NAME IS OH!FM." THEN 100 ELSE 200
30 END
100 PRINT "You are my friends!!!" : END
200 PRINT "You are my enemies..." : END
```

リスト 5

```
10 INPUT "How old are you,Baby";ANTANOTOSHI
20  IF ANTANOTOSHI>=20 THEN 100 ELSE 200
30 END
100 PRINT "You are an Oban Gal..." : END
200 PRINT "You are a Genuine Gal!!" : END
```

リスト 6

```
LIST

5 INPUT "Input Gal's border of years";BORDER
10 INPUT "How old are you,Baby";ANTANOTOSHI
20  IF ANTANOTOSHI>=BORDERTHEN 100 ELSE 200
30 END
100 PRINT "You are an Oban Gal..." : END
200 PRINT "You are a Genuine Gal!!" : END

Ready
RUN
Input Gal's border of years? 15
How old are you,baby? 62

Syntax Error In 20
Ready
■
```

リスト 7

```
LIST

5 INPUT "Input Gal's border of years";BORDER
10 INPUT "How old are you,Baby";ANTANOTOSHI
20  IF ANTANOTOSHI>=BORDER THEN 100 ELSE 200
30 END
100 PRINT "You are an Oban Gal..." : END
200 PRINT "You are a Genuine Gal!!" : END

Ready
RUN
Input Gal's border of years? 20
How old are you,baby? 16
You are a Genuine Gal!!

Ready
■
```

IF文中にTHENがなくなってしまっていることにもなるわけ。だから，BORDER　THENと1文字以上，離してあげれば，めでたく虫取り成功と相成るわけなんだ（**リスト7**）。

このエラーは人の作ったプログラムを打ち込むときとか，プログラムの見やすさを無視して圧縮したときなどに，好んで発生する。**リスト8**を見てほしい。これはOh!FM 5号の山本正則氏製作ショートプログラムの抜粋である。1行80文字以内に収めるために，マルチステートメントを多用したり，スペースを省いたりしている点で苦労が忍ばれる。が，190行の行末のIF文に注目。よーく見るとTHENの前に1文字，ELSEの前1文字が，それぞれスペースになっている。200行にも目を移してみよう。中ほどからのIF文中では，スペースは1個しかない（このスペースはなくてもかまわないけど）。そう，THENやELSEの前は数値（定数）だから。初めの方でｉ）やii）のからみがあると言ったことは，実は今説明してきたような場合が多いからなんだね。ｉ）やii）のミスが，本当に気をつけなければいけない，いや絶対に忘れてはいけない大切なキマリであることがこれでわかってもらえたと思う。

vi）はケアレスミスに近いもの。たとえば，

```
10 DEF FNA(A,B,C)=B*B-4*A*C
20 INPUT A,B,C
30 PRINT FNA(A,B,C)
```

というプログラムは，判別式を表したもの。$AX^2+BX+C=0$（ただし$A\neq0$）なんて方程式の根が実数解か虚数解かを調べるのに使うアレだョ。まぁ，3つくらいを引数にもつ関数を自分で作ったのならば，そう間違いはないはずなのだけれども，引数が増えてきたり，複雑な関数を作ったときに生じやすい虫なのだ！　対策としてはやっぱりプログラム中に注釈を入れるべきだと思う。さっきのもこんな具合にしたらGood！

```
1 '************************************
2 '*                                  *
3 '*  ｺﾚﾊ、 AX^2+BX+C=0                *
4 '*         ﾉ ｶｲ ｦ ﾊﾝﾍﾞﾂｽﾙ Program ﾃﾞｽ｡ *
5 '*                                  *
6 '*     A,B,C――――――ﾋｷｽｳ(3 ﾂ)          *
7 '*     FNA(A,B,C)=B*B-4*A*C          *
8 '*                                  *
9 '************************************
10 DEF FNA(A,B,C)=B*B-4*A*C
20 INPUT A,B,C
30 PRINT FNA(A,B,C)
```

これくらい加えておくとほとんどエラーは起きないハズだし，誰がみてもわかるだろうから，別の人が使ったりしても間違った使い方はしないと思う。やっぱりプログラム作りは，ドキュメンテーションが大切だなと痛感してしまう。Syntax虫は以上で終わり。

リスト 8

```
190  I=I+1:COLOR=(6,U):M=M+1:GOSUB220:FOR S=1TO4:IF S=I THEN C=U ELSE C=0
200  COLOR=(S,C):NEXTS:IF I>3THENI=1:GOTO190 ELSE IF M<200 THEN190 ELSENEXTU
```

2. お次はIlle虫!!

Illegal Function Call，これもよく発生する虫。Ille虫が発生する最大原因の１つに，Data文とData Readの不一致によることが挙げられる。この誤りは実は，Sub虫の発生にも関係しているので，後で詳しく説明しよう。ほかにIlle虫は，パラメータのミスによるのが大部分である。ここで１つ，F-BASICの面白いところを紹介しよう。

```
10 CLS
20 LINE(0,0)-(20000,20000),PSET,7,BF
```

何の変哲もない上のプログラム，7，8ではエラーになる。Ille虫が出るハズだョ。ところが11ではエラーとならない。なぜだろう。理由は次のとおり。7，8のグラフィックは640×200ドットの１モードしかないのに対して，11では640×200と640×400ドットのモードが存在するからだ。「たったこれだけの違いでどうして？」と言われる方がいるかもしれないけど，このたった１つの違いがコンピュータの性格を大きく変えているんだ。

ご存知のとおり，コンピュータ内部で処理できる数値は，１バイトまたは２バイトの数値。そしてそれらおのおのを10進数に当てはめると，255，65535までの整数値である。するとどうだろう，640という数値，正確には0～639までの640個の整数を処理するには，当然65535まで扱える２バイトでなければならない。ところが200であるならば，同様の理由から１バイトで事足りてしまうのだ。そこで7，8のグラフィックルーチン（主にサブCPU）では横方向を２バイトで，縦方向を１バイトで処理している。けれども11では縦が400のモードがあるので，縦横とも２バイトで処理している。だから7，8の場合，下のようにすればエラーは出ない。

```
10 CLS
20 LINE(0,0)-(65535!,255),PSET,7,BF
```

また11なら下のようにしてもエラーは出ない。

```
10 CLS
20 LINE(0,0)-(65535!,65535!),PSET,7,BF
```

ただし7，8ではマイナスの数値は絶対だめなのに対して，11では－65536まで数値は使える。ちょうど７，８が（０，０）を原点とした絶対座標しか使用できないのに対して，11ではかなり広大な範囲の座標系の一部がCRTであると考えられ得ることを考慮しているみたいで，ひょっとしてBASICのバージョンアップ（現在はV4.0が標準で，V5.0なんてのも発表になったけれど）などで広大座標系の設定や，ウィンドウ機能などのグラフィックに便利な命令が加えられたBASICが出るのではないかと期待してしまう。なにしろ，ここら辺のことはマニュアルには載っていないオモシロイ点だと思うナ！

さて話を元に戻して，Ille虫はパラメータを必要とする命令で出る。しかもそれは，パラメータに制限があるものに限られる。だからSINなんて関数は，引数にどんな数値を入れてもエラーになることはないのに対して，SQRなんて関数は「0または正」という制限を破ってマイナスの数値を入れたら，即Ille虫誕生となるわけなんだ。ゆえにIlle虫が出たときには，必ず引数の値を確かめるようにしよう。**リスト３**の40行はIlle虫を出さないための処理である。もし40行がないと，FNAはマイナスとなることがあるわけで（たとえばA，B，Cに１を代入したらFNAは－３）そんなことが起きれば，

```
A,B,C? 1,1,1
KAI1=

Illegal Function Call In 50
```

が出てしまうだろう。こんなときはエラーが出た時点でその行を確認すると同時に，第２章で言ったことをしてみよう。

```
Illegal Function Call In 50
Ready
PRINT B,FNA(A,B,C),2*A
 1          -3            2

Ready
■
```

この結果，確かにFNAがマイナスになってるね？　そしてそれがSQRの引数になっている。SQRの方では許された引数ではないので，Ille虫を出してしまうわけなんだ。Ille虫はこのような手順を踏めば，結構，簡単にとれる虫だから，そんなときはF-BASIC文法書で再確認してみよう。きっといろいろな制限を忘れていた自分に気付くハズ。

3. そしてSub虫!!

まずは**リスト９**を見てほしい。これはOh！FM ２号掲載のタイプ練習プログラムのリストの一部で，別にエラーなんて起きそうもないように見えるけど，実行させると，

```
Subscript Out Of Range In 1580
```

とエラーが出てしまう。このSub虫，どうして出るかご存知？

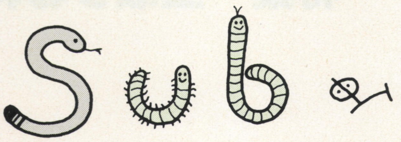
Subscriptとは日本語にすると「予約」という意味になる*5。だからSub虫は「予約の範囲外」の虫なのである。はて，予約といったってピンとこないが……，ごもっとも。この「予約」は予約語の予約とはちょっと違う。後者がBASICの予約であるのに対して，前者はプログラムの予約，なのである。そしてその予約とは「配列の宣言」のことなのである。

「えっ，SYMBOL命令は配列宣言なんかいらないはずだ」まったくそのとおり。正しいSYMBOL命令なら，もちろん予約なんて

リスト９

```
1580 SYMBOL(112,180),"ﾜｸﾉﾅｶﾉ ﾓｼﾞｦ ｳｯﾃｸﾀﾞｻｲ.",2,2
```

*5 Subscript：本当はSubscriptionが「予約」という意味で，Subscriptは「下に書いた文字・記号・数字」という意味。H_2SO_4の２や４がそれに当る。最近のプリンタでは，ほとんどの機種でこれが可能になる。Subscriptを予約と訳したのは，こうしないと意味が通じないから。一種の慣用表現と思ってほしい。ちなみに逆はSuperscriptで「上に書いた～」である。



必要ない。でもさっきのプログラムではSub虫が出たわけで，よく見てみよう。うーん，でも誤ってないみたいだ。困ったもんだ。こういうときはダイレクト命令でできるなら，それでやってみよう。

```
SYMBOL(112,180),"ワク ノ ナカ ノ モジ゛ ヲ ウッテ クダ゛サイ。",2,2
```

ほら，ちゃんと動くじゃないか。でも絶対エラーになってしまう，さっきのプログラム……。そうだ！　プリンタでリストを取ってみよう。どれどれ，うーん，さっぱりわからん。どう見ても間違いはないよね。でも実はあるんだ。下のリストを見てごらん。

```
1580 SYMB0L (112,180),"ワクノナカノ モジ゛ヲ ウッテクダ゛サイ.",2,2

1580 SYMBOL (112,180),"ワクノナカノ モジ゛ヲ ウッテクダ゛サイ.",2,2
```

一目瞭然。そう「O（オー）」と「Ø（ゼロ）」の違いなんだ。プリンタを替えただけでわかってしまったこのミス，実は結構多いんだョ。たった1文字のミスなんだけど，コンピュータは認めてくれるわけがない。頑固だからネ。まあ，正しい指使いでキーをたたいているならば，ほとんど出ないミスだけど，コンピュータのキーボードはユーザのニーズによる多様化，機能の複合化などによりキートップにはいろいろな文字，記号が書かれており，その使い方もかなり複雑になっている*[6]。と同時に，逆に初心者にとっては難しいものとなってしまっている。でも，オーやゼロはキーボード上ではちゃんと別々に書いてある。FMシリーズではオーを「Ơ」，ゼロを「0」と書いているし，CRT上ではオーは「O」，ゼロは「0」となっている。だけどうっかり見落としてしまう。じゃあどうするか，というとその解決法は2通りある。

1つは，プリンタ内部のディップスイッチで，ゼロにスラッシュ（斜め線のこと）を入れたり入れなかったりできるものは，スラッシュの入るモードに変えてやること。これはプリンタによって違うから，できないものもある*[7]。その場合は次の方法が良いと思う。すなわち，ハードコピーを使ってやる手段なんだ！　エラーの出た行まわりのリストをとって，ハードコピーする。その場合，HARDC1かHARDC2でハードコピーをとればゼロにスラッシュが入る。少々面倒くさいけど，以上のどちらかをやってみることを，Sub虫が出たとき勧めるネ。プリンタを持っていない人は，2人ぐらいでリストを読み合せすることを勧めます。

4．さらにRead-Dataエラー

編集室にくる質問電話でも，これは横綱格。たとえばOh！FM 3号の僕のプログラム「中間色PAINTもどきプログラム」（FILES Oh！FMでは227ページ）のうしろの方のDATA文のエラーがとても多かった（3号が出たとき，虫が出ちゃう人多いと思ってた。事実，機械語のエラーはほとんどなくて，DATA文のエラーの電話が多かったことを覚えている）。とくに多かったのが，680行の2行目で，72.5のところを72,5と2つのデータにしてしまった，というものだった。するとどうなるかというと，このデータは「モモちゃん」の絵のデータで，大部分次のフォーマット 0，X座標，Y座標～(以下このペアの繰り返し～)0，−1(ここで終わり)だったものだから，1つのデータがずれてしまってデータエンドであることがわからず，しかも，LINE文中に−1の数値を取り込んでしまうのである。そうなれば，Ille虫が出ちゃうことはさっき述べたことからおわかりいただけるだろう。だからFILES Oh！FMでは親切に，「ピリオドだゾ!!」と注意書きを入れたんだヨ（FILES Oh！FMは1，2，3号をただ集めて合本にしたのではなく，虫という虫は徹底的に殺し，かつ発生させないようにした，これはもうスゴイ本なんだヨ！）。でも，どうもピリオドとカンマの違いによるミスって奴は多いようだ。とくにDataが数値の場合，

1. 見にくいリスト
2. たまにピリオドが出る
3. いろいろな桁数が入り混じっている

なんてリストを打ち込むとき，ミスをしやすい。誌面の都合を考えてプログラムリストを載せてしまったために，1．のようなリストとなって結局ミスを誘発してしまったみたい。（ここで言いわけを一言：もとのプログラムは実に見やすかったんだゾ。だって，1行にDataを2個しか入れなかったんだもん！　でもそれでは，ズラズーラのリストになってしまうから，Data文のところだけギッシリつめてしまったんだよ。これ，ホント!?）。2．なんかも気をつけようね。3．も同じ。だから，皆さんがData文の多いプログラムを作るときには，1，2，3，をまとめて気をつけるようにしよう。そのためには，

1. DATA文だけある程度まとめる
2. RESTORE命令を使うようにしよう
3. REM文を使って，DATAのフォーマットを書いておこう
4. リストをとったとき，カンマの位置がそろうようにしよう
5. DATAの個数を明記しておこう

などに注意しよう。ちょうど，機械語のDATAをBASICのプログラム中に入れて，FOR～NEXTループとPOKE文でメモリ上に展開する際のようにするわけ。たとえば**リスト10**と**11**。どちらがいいと思う？　**リスト10**の方が絶対読みやすい。ただし機械語をDataにしているときには，ピリオドが出てこないから，ピリオドの入れ間違えというのは起こりにくいと思うネ。

そのほかDATA文の終わりを知るのに，

1. ENDマークを置く

*[6] 8年ほど前，あの有名なアップルが出たとき「タイプライタみたいだ！」と思えるぐらい，キー配列，キートップは簡単なものだったんだョ。だいたいがひらがな，カタカナだけで50種類以上，JIS第一水準の漢字だけで3000以上の日本語を全部キーにするなんて，無理な話。だけどせめてカタカナぐらいはキーにのせようってことで，現在あるような形に落ち着いた。カタカナだとCRTやプリンタへの表示の際のデザインが楽なんだ。日本人だもの，日本語使いたいよね。

*[7] ビジネスプリンタにはスラッシュの入らないものが大部分。

　2．あらかじめ特定個数だけREADするようにする

などの方法がある。だけど1．だと「モモちゃん」のエラーみたいにIlle虫が出てしまうし，2．にいたっては実際のDATAの個数とREADするときの違いによって，Ille虫やらSyntax虫，Sub虫まで発生する可能性がある。だから，

　1．いろいろなデータが混在しているときは，ENDマークをいろいろ変えてみる

　2．DATA文の先頭にDATAの個数を入れておく

　3．DATAの個数，種類などをまとめて別のDATA文とするのもよい。どんなやりかたでも，皆さんがわかりやすいようにするのがベストです。それでもエラーが出る場合は，

　1．READ文の部分のみダイレクトモードでやってみる

　2．READしたパラメータを表示させてみる

という方法がある。これは何度も言ってきたことなので，もう充分理解していただいていると思う。たとえば先ほどの「モモちゃん」のRead文のところ（**リスト12**）を**リスト13**のようにプリントさせてみる。あまり手を加えてしまうと，また変てこリンなエラーを出してしまうよ。もちろん**リスト14**のように見やすくしてあげてもよい。このようにすればAやB，CやDにマイナスの数値が入ってはいけないことから，ピリオドなしによるDataの変化はわかるはずだ。エラーが出て，Dataがあるときは，こんなふうに虫をやっつけちゃおう。

5．エラーメッセージが出ないエラー

　今までは発生件数の多いものを中心に話してきましたがエラーは，ひょんなところにも潜んでいることがある。一見，プログラムはエラー表示をして止まることがないので正常そうなんだけど……。下のリストを見てほしい。

```
1000 IN$=INKEY$
1010 IF IN$="" THEN 1000
1020 PRINT IN$
```

　見てわかるように，1文字入力のプログラムで，INKEY$を利用したやつ。これが下のリストみたいだったらどうなると思う？

```
1000 IN$=INKEY$
1010 IF IN$=" " THEN 1000
1020 PRINT IN$
```

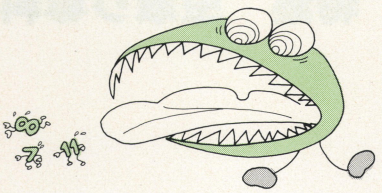
　そうです，ちっとも1文字入力になっていない。違いはわかるネ，そう，ダブルクォーテーションのところ。BASICの基本的項目で「文字定数」というのがあるよね。これははじめのうちはな

リスト10

```
10000 '* MACHINE LANGUAGE POKE ROUTINE *
10010 CLEAR,&H4FFF
10020 MA=&H5000
10030 FOR I=MA TO MA+237
10040   READ A$:A=VAL("&H"+A$)
10050   POKE I,A
10060 NEXT
10070 RETURN
10080 '* MACHINE LANGUAGE DATA *
10090 '     +0 +1 +2 +3 +4 +5 +6 +7
10100 DATA 30,8C,65,AF,8C,54,30,8D
10110 DATA 00,D5,AF,8C,55,CC,C0,00
10120 DATA ED,8C,63,CC,60,00,ED,8C
10130 DATA 3D,AE,8C,3A,31,8C,5C,C6
10140 DATA 64,A6,80,A7,A0,5A,26,F9
10150 DATA 30,8C,2D,AD,FF,FB,FA,EC
10160 DATA 8C,44,C3,00,64,ED,8C,3E
10170 DATA EC,8C,1B,C3,00,64,ED,8C
10180 DATA 15,EC,8C,10,83,00,64,ED
10190 DATA 8C,0A,2A,CD,30,8C,11,AD
10200 DATA FF,FB,FA,39,00,00,00,00
10210 DATA 10,00,00,00,00,77,00,00
10220 DATA 10,00,00,00,00,0F,00,00
10230 DATA 00,00,3F,59,41,4D,41,55
10240 DATA 43,48,49,91,D3,93,00,00
10250 DATA 00,64,90,00,00,00,00,00
10260 DATA 00,00,00,00,00,00,00,00
10270 DATA 00,00,00,00,00,00,00,00
10280 DATA 00,00,00,00,00,00,00,00
10290 DATA 00,00,00,00,00,00,00,00
10300 DATA 00,00,00,00,00,00,00,00
10310 DATA 00,00,00,00,00,00,00,00
10320 DATA 00,00,00,00,00,00,00,00
10330 DATA 00,00,00,00,00,00,00,00
10340 DATA 00,00,00,00,00,00,00,00
10350 DATA 00,00,00,00,00,00,00,00
10360 DATA 00,00,00,00,00,00,00,00
10370 DATA 00,00,00,00,00,00,00,00
10380 DATA 00,3F,59,41,4D,41,55,43
10390 DATA 48,49,93,C0,00,90
```

リスト11

```
10000 '* MACHINE LANGUAGE POKE ROUTINE *
10010 CLEAR,&H4FFF
10020 MA=&H5000
10030 FOR I=MA TO MA+237
10040 READ A$:A=VAL("&H"+A$)
10050 POKE I,A
10060 NEXT
10070 RETURN
10080 '* MACHINE LANGUAGE DATA *
10090 DATA 30,8C,65,AF,8C,54,30,8D,0,D5,AF,8C,55,CC,C0,0,ED,8C,63,CC,60,0,ED
10100 DATA 8C,3D,AE,8C,3A,31,8C,5C,C6,64,A6,80,A7,A0,5A,26,F9,30,8C,2D,AD,FF
10110 DATA FB,FA,EC,8C,44,C3,0,64,ED,8C,3E,EC,8C,1B,C3,0,64,ED,8C,15,EC,8C,10
10120 DATA 83,0,64,ED,8C,A,2A,CD,30,8C,11,AD,FF,FB,FA,39,0,0,0,0,10,0,0,0,0
10130 DATA 77,0,0,10,0,0,0,0,F,0,0,0,0,3F,59,41,4D,41,55,43,48,49,91,D3,93,0
10140 DATA 0,0,64,90,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10170 DATA 0,0,0,0,0,0,0,0,3F,59,41,4D,41,55,43,48,49,93,C0,0,90
```



かなかわからないもので，とくに「空文字列」とか「ヌルストリングス」なんてやつ。早い話が普通の数値の「0」にあたるものだけど，それが「文字定数」では，ダブルクォーテーションを続けて2つ並べたものなんだ。INKEY$というBASICのステートメントは，「キーボードが押されていればその文字を与え，押されていなければ空文字を与えます」なんだ。だから，

「押されるまで待つ」

ことがINKEY$を使った1文字入力では必要であるわけで，そのためには，別の言い方をすれば，

「空文字の間は待つ」

ということになる。だから先ほどのプログラムのようになるわけで，誤ったプログラムでは，

「空文字ではなくスペースキーが押されるまで待つ」みたいになってしまっているので，空文字を待っていたINKEY$はIF文中で条件を満たさないため，あっという間に次の行へと行ってしまうわけだ。だから，「カーソルみたいのが出るけれど，キーを受け付けてくれない」とか，

「キーを入力してくださいと言ってくるけど，キーを入力しないのにどんどん次の処理をしてしまう。バグと違いますか？」

なんて編集室に電話してくる人の大部分は，こんなミスをしているようだ。ある程度慣れてくれば，一種の定石的な使い方をするのでわかるんだけど，慣れないうちはどうもこんなエラーが多いようである。

このほかに，エラーメッセージが出ないとか，一見正しそうなプログラムが思いどおりに動かない場合の大部分は，アルゴリズム――すなわち人間がコンピュータに仕事をさせる際にその手順を考えること――における誤ちであると言えよう。プログラミングの基本中の基本である，と断言してもいいくらいにアルゴリズムは大切なことで，これがしっかりとしていないと，正しそうな（エラーが出ないという意味）プログラムでもまったく役立たずのものを作ってしまう危険性があるんだ。ゲームプログラムを作るのもよし，ビジネスプログラムを作るのもよし，何でもいいんだけど，プログラムを作るとき，自分のやらせたいことをコンピュータに具体的に，かつ正確に教えてあげられるようにならなくては，この根本的な虫は，いつもあなたに付いてまわることになる。だから，物事を細分化する考え方や，具体化する実例などを覚えることが，虫のいないプログラム作りや，プログラム中の虫を想像できるようになる一番の近道だと思えるんだ。やはり何ごとも，一朝一夕にはできないものだと言えるようだネ。

6. リスト打ち込みで発生する虫のお話

これもまたとても多いミスの1つ。とくに次のような文字や記号で発生しやすい虫である。

❶ O（アルファベットのオー）と 0（数字のゼロ）

Sub 虫のところでも述べたとおり，とくに気を付けてもらいたいもの。本誌でもこのところに気を配って，必ずリストにはスラッシュの入ったゼロを使用している。それでも結構間違える人が多いので，注意すること。また，スラッシュの入らないプリンタでデバッグするときは，よく見て字体の違いに注意しよう。オーの入っているBASICのステートメントでは，とくによく見よう。オーの入るBASICの命令には，

```
AUTO   LOAD   MON    FOR     GOTO  STOP
POKE   ERROR  MOTOR  TRON    CONSOLE
COLOR  LOCATE CLOSE  CONNECT
SYMBOL EOF    LOF
```

などがある。ステートメントの前後関係により，ゼロでないかチェックしよう。

❷ （アルファベットのビー）と （数字のハチ）

機械語のダンプリスト打ち込み時のエラーは，大部分がこれ。ビーのところをハチにしてしまって見事暴走，チェックサムの見間違いであえなくRun away。BASICでもビーの入る命令でのエラーはこれがほとんどで，ビーの入る命令は下のとおり。

```
GOSUB  BEEP  SYMBOL ABS  CDBL  TAB
```

❸ I（アルファベットのアイ）と 1（数字のイチ）

これも間違えやすい。機械語入力のときに間違える人はいないと思うけれど，英文タイプライタに慣れてる人では，小文字のエ

リスト12

```
410 READ E:IF E<0 THEN RETURN
420 READ A,B:LINE(A*2.3,B)-(A*2.3,B),PSET,E
430 READ C:IF C=0 THEN 410
440 READ D
450 LINE -(C*2.3,D),PSET,E
460 GOTO 430
```

リスト13

```
410 READ E:PRINT E 'IF E<0 THEN RETURN
420 READ A,B:PRINT A,B 'LINE(A*2.3,B)-(A*2.3,B),PSET,E
430 READ C:PRINT C 'IF C=0 THEN 410
440 READ D:PRINT D
450 'LINE -(C*2.3,D),PSET,E
460 GOTO 430
```

リスト14

```
410 READ E:PRINT "E=";E 'IF E<0 THEN RETURN
420 READ A,B:PRINT "A,B=";A;",";B 'LINE(A*2.3,B)-(A*2.3,B),PSET,E
430 READ C:PRINT "C=";C 'IF C=0 THEN 410
440 READ D:PRINT "D=";D
450 'LINE -(C*2.3,D),PSET,E
460 GOTO 430
```

ルがタイプライタではイチでもあるので，間違えていることが多いみたい。でもBASICの方で大文字に直してリストしてくれるので，リストをよく見ようネ（ただしPRINT文中のダブルクォーテーションで囲まれた中の小文字のエルはなおりません）。

以下にアイの入るBASICの主なステートメントを示そう。

```
LIST    FILES   KILL    DEFINT IF       DIM
CHAIN   INPUT   PRINT   LINE    PAINT   TIME
SIN     MID$    INPUT$  INKEY$
```

❹ ,（カンマ）と .（ピリオド）

記号で間違えやすいのはこれ。カンマは

ⅰ）パラメータの区切り

ⅱ）データの区切り

に使われている。ⅰ）の例には，

```
WIDTH 80,25
LINE (0,0)-(639,199),PRESET,,BF
LOCATE 10,10
COLOR 1,7
```

などがある。ⅱ）の例には，

```
DATA 123,456,789,987,654,321
```

がある。それ以外にはPRINT文中で使うくらい。ピリオドは

ⅰ）小数点

ⅱ）省略命令

でよく使われる。ⅰ）の例として

```
DATA 160,.5,.76,90,659,.263,94,.2
```

などがそうで，とくにカンマと近いときには気を付けよう。また，わざわざゼロを打ち込んだり，空文字を打ち込むのが面倒な場合にカンマを続けていることがある。たとえば，

```
DATA THIS,,IS,,A,,PEN,.
CIRCLE (200,150),100,3,.3,,,F
CONSOLE ,,,1
```

などがそうで，こんなときにピリオドが混じっているとまどわされかねないので，充分注意すること。ⅱ）の例としては，

```
W. 80,20   L.     L..    L...    C.    LO.
LO.M  RET.  COL.   R.     CONS.   ME.   GO.
GOS.  LOC.
```

がそう。LO.Mはマニュアルで説明してないけど，

```
LOADM
```

のこと。ピリオドはこんなふうに使われている。PRINT文中にも出てくるけど，それはあまり神経質になる必要はないんでないかな。

❺ ;（セミコロン）と :（コロン）

セミコロンは，PRINT文やINPUT文中で使われ，一方のコロンは，ステートメントを区切るときに使われている。BASIC中ではそういうきまりになっているので，このまま覚えてしまおうね。BASIC以外の言語では，逆の働きをしたりするものがあるけれど，それらはBASICをマスタしてから覚えても遅くはないから，そうあせらずに，一歩一歩ゆっくり確実にものにしていくようにしよう。

❻ -（マイナス）と -（カナの音引き）

マイナスは，

```
LINE (4,4)-(100,100),PSET,1,B
A=A-2
```

などに使われており，テンキーやフルキーのØの右横にあるキーを押すと出てくるやつ。音引きは，カナモードで，フルキーのバックスペースキーの左横にあるキー。とてもよく似ているので，間違うことが多いようだけど，コンピュータは許してはくれない。

マイナスはいろいろなところに顔を出すけれど，音引きは，ダブルクォーテーションの中だけ。

❼ =（イコール）とニ（カナのニ）

ときどき見られるミスで，よく見れば違うことに気付くはず。とくにカナキーのランプがついている状態かどうか，などに注意してほしい。これと似た誤りで，スラッシュと，グラフィックの斜線をまちがえる人がいるけれど，こちらもグラフィックキーは押さない，という点にもう一度着目すべきだろう。

以上の注意点を念頭において入力，デバッグすれば，かなり長いプログラムでも一発でうまく打ち込めてしまうと思う。しつこく「覚えてほしい」なんて言ったけれど，無理に覚える必要はありません。忘れてしまってもかまいませんが，BASICの文法書を座右の銘にして，一生懸命読んであげようね。きれいに飾っておくより，ボロボロになるくらい，何度も読んで，確実にものにしていくようにしよう。

他に，大したことではないんだけど，ダブルクォーテーション内のスペースの数によって，プログラム上，何の支障もないのだけれども，CRT上でうまくいかないことがよく起るようだ。すなわち，WIDTHが40字のときに40字以上の文字列を，80字のときに80字以上の文字列を画面に出力した際，BASICの方では改行してしまって，CRT上でおかしなことが起こってしまうわけなんだね。文字列が長すぎて，画面がスクロールアップしてしまったり，逆に短くて次の文がくっついてしまったりなんてことが起きるようだ。ここらのことは非常に親切なFMシリーズのマニュアルにも書いてないので，心配になった初心者と思われる方からときおり質問がくるのですが，根本的にはリストをそのまま，正しく打ち込んで，としか答えられません。けれども実際には**リスト15**のような場合，なかなかスペースの数まで間違えずに打ち込むのは難しいようだ。そこで1つの解決策として，昔，僕が初心者のころ考えた，スペース打ち込み法を紹介しよう。

7. スペースよ，おまえはいくつ?

僕は次の方法でスペースの数を解決していた。

1. リストのコピーをとり，縦に線を引く
2. リストに合わせたスケールを作る

「なーんだそんなこと」なんて言わないで。これでも結構，マジに悩んだ時期があったんだから。

7.

1．も2．も簡単にできる。1．はまず，コピー機械の置いてある本屋さんか文房具屋さんに行って2枚ぐらいリストのコピーを取る(本屋さんで取る場合は本を買ってからね)。そして家に帰ったら長い定規で，縦線を引くんだ。何も1文字ずつ引く必要はなくて，5文字とか10文字ぐらいずつでかまわない。ここまでできればあとは簡単。罫線に合わせて，数を数えればいいのだから。

2の方法もたいしたことじゃないよね。ちょっと厚めの紙を定規のように短冊に切ってから，本のリストに合わせて区切りの線を入れてやるわけ。これも前のと同様，1文字1文字すべて線を引く必要はない。5文字目に1本，10文字目は少し長めに，なんてことをしながら線を引けば，効率が良いと思う。もちろん，リストの幅，めいっぱいにこれをやってもかまわないけど（幅はたかだか80字だから)，使用する人が各自，好きな長さでやってみよう。こうすればPRINT文中のスペースは，あっという間に入力できる。

8．忘れ物，忘れ物……

話が前後してしまったけれど，Ille虫での忘れものを1つ。RANDOMIZEって命令も，Ille虫の出やすい原因の1つである。ゲームの先頭でよく使う命令だけれども，この後ろにつけるSeed Numberに制限があるためにIlle虫の発生を起こすことがあるんだ。ご承知のようにこの命令は，乱数の系列を変えるためにある。でもSeed Numberが同じものだと，乱数の系列は結局いつも同じことになってしまう。そこで，

```
RANDOMIZE TIME
```

なんていうふうにSeed Numberに予約変数の1つ，TIMEをあててあげると，プログラムをRUNさせるごとに乱数の系列はかなり変わってくるんだね。でも「Seed Numberは—32768から32767までの間の数値ではなくてはならない」という制限があるために，TIMEの値がこの制限外だとIlle虫の誕生となるわけ。何人ぐらいの人がやっているかは不明だけど，コンピュータの電源を入れたときに，TIME$に現在時刻を入れたりする場合に，このエラーを発生させてしまうようだ。TIMEは知ってのとおり，0時から現在までの時間の秒数を持つ変数だから，TIME$が9時6分7秒を超えてしまうと，TIMEはもうSeed Numberとして使えなくなってしまうのである。

```
LIST

10 INPUT IN$ : TIME$=IN$
20 RANDOMIZE TIME
30 GOTO 10

Ready
RUN
? "09:06:07"
? "09:06:08"

Overflow In 20
Ready
■
```

だから，

```
RANDOMIZE TIME/3
```

としてやれば，エラーは出なくなる。というのもTIMEは0から86399までの整数値なので，3で割ってやれば0から28799までの整数値となり，Seed Numberの条件を満たしてくれるから。これが面倒だという人は，スイッチを入れてすぐにそんなゲームをやってしまえばいい。そうすれば3で割ったりする必要はなくなってしまうから！

さて次は，Sub虫の忘れもの。**リスト16**を見てくれる？

これは本GO君が3号で作った，本誌用チェックサムプログラムの抜粋。エラーが潜んでいるんだけど，おわかりかな？　そうCHR$をCHRとして$を抜いてしまっているところがあるでしょう。これでめでたくSub虫の誕生なのだった。Sub虫の章で述べたように，SYMBOL文中のオーをゼロに間違えたため，BASICがSYMBØLという変数名と間違え，さらにカッコがついているので配列だと思ってしまったのと同じミスなんだ。CHR$のような関数は，$まで入って初めて文字関数の1つだとBASICが理解するわけで，$がないと，CHRという変数，配列だと思ってしまう。そのうえ，宣言外の引数を持っていたため，Sub虫を発生させたわけなのである。

ご存知のごとく，配列の宣言は10以内の1次元の配列なら，宣言しなくてもかまわないので，この辺は注意が必要だ。小さな引数だったら見逃してしまう，エラーの出ないエラーになる可能性をも秘めている。

以上のように，関数やカッコによって引数が必要な命令でのSub虫は，命令語をしっかりフルネームで書いていないのが発生原因の大部分となるので，Sub虫が出た行では，この辺に注意してよく見てみよう。

リスト15

```
10 WIDTH 80,25 : CONSOLE 0,25,0,0 : COLOR 7,0: CLS
20 PRINT "┌──────────────────────────────────────────────┐"
30 PRINT "│        コノ PROGRAM ハ デモンストレーション ヨウ ノ モノデス。        │"
40 PRINT "└──────────────────────────────────────────────┘"
```

リスト16

```
630 IF (I$<"0" AND (I$<>CHR$(13) AND I$<>CHR(28) AND I$<>CHR$(29) AND I$<>CHR$(3
2))) OR (I$>"9" AND I$<"A") OR I$>"F" THEN BEEP:GOTO 610
```

もう1つ，忘れてはならないエラー命令に，ON ERROR 命令がある。これは，ON GOTO 命令とよく似ている。ON GOTO 命令が，引数に変数を入れて，その値によっていろいろと分岐するのに対して，ON ERROR 命令は「エラーが起こったら分岐しろ」という命令なのである。ということはつまり，プログラム中でエラーが起こった場合に，ある処理ルーチンでエラーの原因を明らかにしたり，そのエラーから，元のルーチンへ復帰してプログラムを中断することなく実行することが可能となるわけ。そこで必要となってくるのが，前に説明した予約変数，ERR と ERL なんだ。これによって，あらかじめ予想されるエラー（たいていは人為的なエラーで，プログラム上のエラーでない点が重要だけど）に対して，手を打っておくことができるんだねェ。とくにこの命令は，実務用のプログラムでは重宝しているもので，たとえば，ディスクを使用する（ファイルをフロッピ上に作ったり，読み込んだりする）ものでは，フロッピがディスクに入っていない場合や，フロッピの入れ間違いによって使用するファイルがフロッピ上にない場合なんかがそれで，これは使用する人によって起こりうるエラーなんである。でも，人為的ミスでいちいちプログラムが中断されてはたまんないよね？　逆に，そんなミスぐらい直してくれたり，人間側に教えてほしい！　と，思うのが人情でしょう。そんなとき，この命令を利用することによって，すべてが丸く納まるのだ！　（まったく，気くばりってヤツは厄介だ。けれど，それを怠ると，とんでもないエラーなどでプログラムがうまく走らない。我慢，我慢）そんな一例を，**リスト17**に挙げてみよう。

これは，Oh！FM 7号の「蓉子の住所録」の最後の部分で，エラー処理をしているところ。16020行で，ERR＝70，すなわち，Bad Record Number というエラーになったとき，“ADD. MAP”を再ロードしてメニューに戻る処理をしている。このプログラムでは，メモリ上にデータを展開して，それを元にいろいろな仕事をしている。でも，もしこのメモリエリアに何か異常（使用者が書き換えてしまったりなど）が起きた場合，コンピュータは仕事ができなくなってしまう。そんなとき，もう一度フロッピよりロードして，仕事に復帰させるための処理をしているんだ（そして RESUME2000 によって2000行へジャンプする）。16030行も同様。ここでは ERR＝63かつ ERL＝3070，File Not Found のエラーが3070行で起こった場合にのみ，エラー処理をしている。たとえば初めてこのプログラムを使うときには，“ADD. MAP”というファイルは存在しないのに，KILL 命令を使用してるネ（**リスト18**）。ファイルがないのに KILL しようとすると File Not Found のエラーとなってしまう。だからその場合は，16030行にてそのエラーを無視して，次の SAVEM を実行して無事に終わるわけ（RESUME NEXT がそれ）。一度でもこのプログラムを実行してから3070行にくる分にはそのとおりの処理をしてくれるので，16030行にくることはない。そして16040行以降では16020行や16030行以外のエラー（Syntax 虫など）が起こった場合，つまり予想外のエラーになるわけだ。そんなときに限って起こったエラーのエラーコードと行番号を表示して，一時処理を中断するようにしているわけ。もしこんなふうにしてプログラムが止まった場合，それはリストどおりではない可能性があるので，今一度デバッグしてほしい，という作者の優しい心使いによるプログラミング例と申せましょう（さすが蓉子ちゃんは女らしい）。だからもう一度リストを文法書などで確認してもらえば，虫はすべてつぶせるはず。頑張ってみよう。

第4章　エラーの出ないプログラムを作ろう！

今まで述べてきたことはすべて，虫が出てからいかにその虫を取り除くか，に重点を置いてきたね。駆除法だったわけだけど，これから述べることはもっと以前の話，そう，予防法について話してみたいと思う。

本誌でもとりあげているプログラムの構造化などは，その1つの例である。その詳しい内容，解説はここでは省くが，なぜその方法が予防策になるかに絞ろうと思う。

構造化プログラミングでは，

1. プログラムが見やすい
2. 何をやっているのか，他人でもわかりやすい
3. 小さなプログラム（サブルーチン）の集合体にできる

などの利点があって，それぞれみな，虫の発生が起こりにくい要

リスト17

```
16000 ´♥ Procedure ERROR ♥
16010   WIDTH 40,25 : BEEP
16020    IF ERR=70 THEN LOADM "ADD.MAP" : RESUME 2000
16030    IF ERR=63 AND ERL=3070 THEN RESUME NEXT
16040   COLOR 2 : PRINT "ｷﾞｮｳﾊﾞﾝｺﾞｳ ";ERL;"ﾆ, ｴﾗｰ ｺｰﾄﾞ ";ERR;" ﾉ ERROR ﾊｯｾｲ!!"
16050   COLOR 4 : PRINT "ｼｮﾘｦ ｲﾁｼﾞ STOP ｼﾏｽ｡ "
16060   END
```

リスト18

```
3070   POKE MTOP,0 : POKE MTOP+1,0 : POKE MOF,EMAK : KILL "ADD.MAP" : SAVEM "ADD
.MAP",MTOP,MOF,0
```

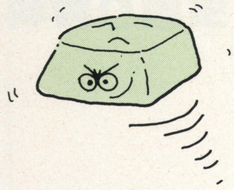


表1　代表的エラーの傾向と対策

エ　ラ　ー		傾　　　向	対　　　策
打ち込みミスによるもの	Syntax Error	○打ち込みミスによるもの ○変数名を間違えている場合 ○その他いろいろ!!	○文法書をよく読む ○プログラム中の変数名を点検
	Illegal Function Call	○Data文，Data-Readの不一致により発生する ○パラメータ，予想外の値によるもの	○Data文の点検(間違いやすい文字に要注意！) ○変数の値を確かめる
	Subscript Out Of Range	○配列宣言のミス ○CHR$やSYMBOLなどのミス	○配列の確認 ○(　)のつく命令の再確認
とりたててエラーは出ないがどこかおかしい (予想通りの値にならない……etc)		アルゴリズムの誤りによるもので，概して他人に発見される場合が多い。人間ならではのミスといえる。	根本から考え直す（コンピュータに融通性はないのです）

因の１つである。１について言えば，見やすいということが，誤りを見つけやすいことに通じるからだと思う。だから，製作中において，ミスは簡単に見つけて直すことができるんだ。変数名などを，わかりやすく，自分でよくわかっている名前にしたりすることもその１つだろうね。いつもいつも，ＡとかＢだけだと，大きなプログラムを作るときに苦労することは，目に見えている。だから，プログラム作りに時間をかけてやることで，虫取りの時間を短くしたり，省いたりすることができる。「急がば回れ！」の精神だね。

２.についても，１.に近いことで，やはりただ漫然とプログラムを作ったり，リストを打ち込んでいるだけでは，はっきり言って虫の発生する余地を残してしまうと思う。１行１行理解できれば，おのずとミスは少なくなるはず。よく，物事がわかっているかどうかを知るには，知らない人に説明することが良いと聞くよネ，このことと似ていると思うんだ。わかったつもりだけでは，必ずミスを招いてしまう。けれども，ほかの人が見て一目でわかるプログラムというのは，こんなミスの入る余地はほとんどないといえる。さもないと，いつまでたっても虫との追っかけっこ，になりかねないからネ！　心掛けよう。

３.は，プログラミングの基本的なこと。いきなり大きなものを作ることは，砂上の楼郭(ろうかく)になることは明らかだ。小さな基本的な部分で地盤を固めておけば，小さな家だろうと超高層ビルだろうとビクともしないはずなんだ。機械語のプログラムの紹介記事にもよく書かれているでしょう？　小さく確実に動いて，虫のいないルーチンができて初めて，何Ｋもの大きなプログラムを作ることができるって。BASICでも同じことで，やはり小さな，数行の確実に動く虫のいないルーチンが存在して，初めて何百行ものプログラムが可能となるんだネ。

とまあ，文章にしてしまうと簡単だけど，なかなかこうはいかないのが現実でしょう。けれどもあまり書きすぎてもよくないと思う。何もかも書いてしまうと，この文章を読んだだけでわかったような気になってしまう，と思うから。やはり少しは苦労してもらって，考えてもらって自分で悩んでもらわないと，こういうことは理解してもらえないだろう。同じFMを愛する者としての少しイジワルな愛のムチ，と取っていただけたら幸い。

第5章　エピローグ

虫の発見，駆除，予防ということについて，かなり飛ばして話をしてきて……。まとまりがなく，かつすべてを網羅しているわけではなかったので，少し不安が残る。でもキーポイントは述べたつもり。失敗を恐れていては人間は前進がないものなんだ！どんどん虫を発生させてもかまわないから，プログラムを作ってみよう，リストを打ち込んでみよう。わからないことはマニュアルを見るなり文法書を読むなりして，１つ１つ確実に自分のものにしていこう。

最後に，代表的エラーの傾向と対策を**表1**にまとめた。本文中ではあちこち飛んじゃってたからね。「プログラムは１日にして成らず」だョ。そしていつの日か，立派なプログラムを投稿してください。待ってるヨ！

特集 虫取り専科

for FM-7,8,11

BASICプログラムのチェックサム

園田 茂

デバッグ特集が決まったとき，編集部内よりBASICプログラムのチェックサムが提案された。PRINT文中のグラフィックキャラクタなど，リストのみでは判断のつかないことも多い。マシン語ばかりではなく,BASICにチェックサムがあってもいいじゃないか，というわけです。そんな折，実にタイムリに園田氏より「BASICチェックサム」プログラムが送られてきました。そこで，園田氏のBASICチェックサムをベースに，一部仕様を変更したものが，今回紹介するプログラムです。

1. 仕 様

一口に言うと，ASCIIセーブされたプログラムの各行のサムを求めるプログラムです。

すなわち，各文字のASCIIコードを加えた値を，マシン語のチェックサムの形でまとめる，というものです。

ただ，少し融通が効くように，

①REM文（´も）の値は0とする。

②スペース（空白）は加算しない。

③ダブルクォーテーションの中は，スペースを含めすべて加算する。ただし，ダブルクォーテーションそのものは加算しない。

④各行のサムは，16進の下2桁を表示する。

⑤行番号はサムに加えない。

⑥表示は，1行8個，10行を1ブロックとする。

としました。

①は，行番号のすぐあとにREM文があるときは00となります。また

```
1Ø PRINT ´1ギョウカイギョウ
```

のとき，加算されるのはTまでです。

ただし，コロンを使ってマルチステートメントにしたときは，コロンも加算されます。

```
1Ø PRINT :´1ギョウカイギョウ
```

②は，次のようなものは，入力ミスにならず，同じ動作をするためです。

```
1Ø FOR␣I=Ø␣TO␣5
1Ø FORI=ØTO5
```

③は，次のような入力ミスをチェックするためです。

```
1Ø IF INKEY$="" THEN1Ø
1Ø IF INKEY$="␣" THEN1Ø
```

また，ダブルクォーテーションそのものを加算から除外したのは，

```
1Ø PRINT "Oh!FM"
1Ø PRINT "Oh!FM
```

を区別する必要がないからです。

2. 実 行 方 法

チェックしたいBASICプログラムは必ずASCIIセーブされていなければなりません。ASCIIセーブは，

```
LIST "ファイル名"↵
```

または，

```
SAVE "ファイル名",A↵
```

というようにします。

本プログラムを打ち込み（あるいはロードして),RUNさせます。

タイトルが表示されたあと，何秒か待つと，チェックするプログラムのファイルディスクリプタを聞いてきます。

先ほどASCIIセーブしておいたプログラムをディスクユニットかデータレコーダにセットしてください。

データレコーダの場合は，必ずリモート端子を接続します。

ファイルディスクリプタは，デバイス名(CASØ：や1：）を入力するときは，必ず"(ダブルクォーテーション)付きで入力してください。

出力を画面にするか，プリンタにするか聞いてきますから，1か2で答えてください。

すると，リスト2のような形で出力されます。

参考文献

1. FM-8活用研究，1982，p.25，FM-8用チェックサムプログラム，工学社
2. OS-9エディタ・アセンブラ・デバッガ解説書(富士通)

3. 編集部より

今回提案したBASICチェックサムがベストであるとはいえませんが，一応，納得のいくものではないかと思います。とくにテープベースの読者には，ASCIIセーブ／ロードは時間がかかり気の毒とは思いますが，これでかなりの入力ミスが防げるのではないでしょうか。今月号より実験的に，長いBASICプログラムにはこのチェックサムを付けてみますので，ご意見・提案がありましたら，どしどしお寄せください。よりよいBASICチェックサムにしていきたいと思います。

リスト1 BASICチェックサム

© 1984 S. SONODA & S. KAWAMURA

```
1000 ' **** Basic sentence checker for FM ****     by Shigeru S.
1010 ' **** Decoded by S.Kawamura
1020 WIDTH 80,25 : CLEAR 21000,&H6EFF : IX=8 : IY=10 : IXY=IX*IY
1030 DIM M$(IXY),MAP(IXY),A(IY),B(IX),LNUM(IY)
1040 PRINT" Basic sentence checker. " : PRINT : PRINT
1050 GOSUB 1340 : ON ERROR GOTO 1280
1060 PRINT " Ascii-saved file descriptor ?  (ex.  "+CHR$(34)+"1:FM-8"+CHR$(34);
1070 INPUT ")   ";FILM$:OPEN"I",#1,FILM$ : LINE INPUT#1,M$(1)
1080 PRINT " 1: Display  2: Printer   Please choose.  ";
1090 R$=INPUT$(1) : PRINT R$ : IF R$="2" THEN OPEN "O",#6,"LPT0:" ELSE IF R$="1"
 THEN OPEN "O",#6,"SCRN:" ELSE 1090
1100 FOR I=1 TO IY : A(I)=0 : LNUM(I)=0 : NEXT.I : FOR I=1 TO IXY : MAP(I)=0 : N
EXT I
1110 FOR J=1 TO IX : B(J)=0 : NEXT J
1120 FOR I=1 TO IXY : LINE INPUT#1,M$(I) : NEXT I : L=IXY
1130 FOR I=1 TO L
1140 IF I MOD IX =1 THEN LNUM(I¥IX+1)=VAL(M$(I))
1150 Q$=USR(M$(I)) : MAP(I)=ASC(M$(I))
1160 IA=(I-1) ¥ IX +1 : A(IA)=A(IA)+MAP(I) : IB=(I-1) MOD IX +1 : B(IB)=B(IB)+MA
P(I)
1170 NEXT I
1180 PRINT : PRINT : PRINT #6," LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM"
1190 ASUM=0 : FOR I=1 TO IY : ASUM=A(I)+ASUM
1200 PRINT #6,USING"##### ";LNUM(I);
1210 FOR J=1 TO IX : PRINT #6,RIGHT$("0"+HEX$(MAP((I-1)*IX+J)),2);" "; : NEXT J
1220 PRINT #6,":";RIGHT$("0"+HEX$(A(I)),2)
1230 NEXT I : PRINT #6,STRING$(IX*3+9,"-"):PRINT#6,"      ";
1240 ASUM=ASUM MOD 256
1250 FOR J=1 TO IX : PRINT #6,RIGHT$("0"+HEX$(B(J)),2);" "; : NEXT J : PRINT #6,
"/";RIGHT$("0"+HEX$(ASUM),2) : PRINT #6
1260 IF R$="2" THEN IF L=IXY THEN 1100 ELSE GOTO 1320 ELSE IF L<IXY THEN PRINT"
Hit any key to end."; : Q$=INPUT$(1) : GOTO 1320
1270 PRINT " 1: next table   2: end  "; : Q$=INPUT$(1) : PRINT Q$ : IF Q$="1" TH
EN 1100 ELSE IF Q$="2" THEN 1320 ELSE 1270
1280 '** ERROR TRAP AND TREATMENT OF END OF FILE
1290 IF ERR=63 AND ERL=1070 THEN PRINT " File not found. Please re-enter." : RES
UME  1060
1300 IF ERR=54 AND ERL=1120 THEN L=I-1 : IF L>0 THEN RESUME 1130 ELSE 1320
1310 PRINT " ERROR occured:  line=";ERL;"  code=";ERR; : END
1320 CLEAR 300,&H6FFF : CLOSE #6 : CLOSE #1 : END
1330 '** POKE ROUTINE
1340 DEF USR=&H6F00: ADD=&H6F00 : RESTORE 1390
1350 READ DA$ : IF DA$="END" THEN RETURN
1360 POKE ADD,VAL("&H"+DA$) : ADD=ADD+1 : GOTO 1350
1370 '** MACHINE LANGUAGE DATA **
1380 '      +0 +1 +2 +3 +4 +5 +6 +7
1390 DATA 10,AE,01,31,3F,A6,84,27
1400 DATA 7E,F6,6F,99,A6,A5,81,20
1410 DATA 27,07,E1,84,27,71,5C,20
1420 DATA F3,E1,84,27,6A,5C,A6,A5
1430 DATA 81,20,27,F5,81,27,27,5F
1440 DATA F7,6F,99,B1,6F,96,26,1A
1450 DATA E1,84,27,16,5C,A6,A5,B1
1460 DATA 6F,97,26,0E,E1,84,27,0A
1470 DATA 5C,A6,A5,B1,6F,98,26,02
1480 DATA 20,3D,F6,6F,99,A6,A5,81
1490 DATA 20,27,12,81,22,27,1D,81
1500 DATA 3A,27,11,81,27,27,28,BB
1510 DATA 6F,9A,B7,6F,9A,E1,84,27
1520 DATA 1E,5C,20,E1,BB,6F,9A,B7
1530 DATA 6F,9A,20,A5,E1,84,27,0F
1540 DATA 5C,A6,A5,81,22,27,E6,BB
1550 DATA 6F,9A,B7,6F,9A,20,ED,C6
1560 DATA 01,F7,6F,99,F6,6F,9A,E7
1570 DATA 21,5F,F7,6F,9A,39,52,45
1580 DATA 4D,01,00,END, of Data
```

リスト2 本プログラムのチェックサム(出力見本)

```
 LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
 1000 00 00 58 06 D5 AD 98 92 :0A
 1080 C2 EF FA 2A 37 8D A1 2A :64
 1160 97 88 F1 A9 B3 F5 68 22 :EB
 1240 26 7D 11 3C 00 CC 6B 7C :A3
 1320 BA 00 05 5A 3B 00 00 BF :13
 1400 EC AB EE B3 F1 D8 D3 DD :B1
 1480 E7 90 B6 FE 07 DD E8 17 :0E
 1560 08 DB 29 00 00 00 00 00 :0C
    0 00 00 00 00 00 00 00 00 :00
    0 00 00 00 00 00 00 00 00 :00
---------------------------------
      14 0A 26 20 F2 B0 C7 0D /DA
```

ショートプログラム集

メダカ取りゲーム――中田憲吾（海老名市）　for FM-7,8,11

子供のころ，ハンカチでメダカ取りをした経験はだれにでもあることでしょう。ハンカチをメダカの下に動かして，サッと上げるとメダカが入っている。時を忘れ，夢中になっていたころを再びゲームで味わってください。

ハンカチの移動はカーソル移動キーで行います。メダカが入ったときにスペースキーを押しますと，ピッ，ピッという音とともにメダカが取れます。

プログラムのフローチャートは構造化設計に従うNSチャートで作ってみました。独学なので誤りがあるかもしれません。NSチャートで書かれたゲームプログラムの例を知らないので，練習のつもりで作りました。

編集部記：とてもほのぼのとしたプログラムです。田舎の小川での春のメダカ取りや，夏祭りの夜店でやった金魚すくいなど幼い日々を思い出してしまいました。♀パターンはメダカというより，足のはえたオタマジャクシのようにみえるのは私だけでしょうか。動きは変化があっておもしろいと思います。

NSチャートによる構造化設計でのJobの考え方は，この場合BASICに適応しにくいように思われますが，プログラムの構造化への試みはどんどんやってみることをお勧めします。

メダカ取りゲームリスト（FM-7用）

©1984 K. NAKATA

```
1000 ´■  メダ゛カ トリ GAME    84/1/15   By K.Nakata   ■  メイン
1010 U0=270:V0=96:U=2:V=1:X0=270:Y0=96:X1=270:Y1=96:HS=0
1020 GOSUB 1060:´ショキ
1030 GOSUB 1110:´ケーム
1040 LOCATE15,19:INPUT"GAME OVER .....PUSH  Y --->> START ",Y$
1050 IF Y$="Y" OR Y$="y" GOTO 1020 ELSE END
1060 ´ショキ
1070 WIDTH80,25:COLOR4:CLS:PRINT@(6,6),&H2561,&H2540,&H252B,&H3C68,&H246A,&H2532
,&H213D,&H2560
1080 LINE(7,28)-(592,199),PSET,5,B:LOCATE19,1:PRINT"Hi-Score = ";HS
1090 LOCATE27,2:PRINT"ヒキ":LOCATE70,0:PRINT"ノコリ":C=0:TIME$="00:00:00"
1100 RETURN
1110 ´ケ゛ーム
1120 U1=560*RND+10:V1=150*RND+30:U=RND*4+C:V=RND*4+C*.5
1130 T=120-TIME:IF T<0 THEN RETURN 1040
1140 LINE(U0,V0)-(U0+16,V0+16),PSET,0,BF:COLOR5:PRINT@(U1,V1),&H216A
1150 U0=U1:V0=V1:U=U-.2:V=V-.2:U1=U1+U:V1=V1+V
1160 IF U1<8 THEN U1=12:U=-U ELSE IF U1>570 THEN U1=570:U=-U
1170 IF V1<31 THEN V1=33:V=-V ELSE IF V1>180 THEN V1=180:V=-V
1180 LINE(X0,Y0)-(X0+72,Y0+36),PSET,0,B
1190 LINE(X1,Y1)-(X1+72,Y1+36),PSET,2,B:X0=X1:Y0=Y1:X1=X1+X:Y1=Y1+Y
1200 RANDOMIZE TIME:S$=INKEY$:IF S$="" THEN LOCATE75,0:PRINT T:GOTO 1130
1210 IF S$=CHR$(&H1D) AND X1>16 THEN X1=X1-16 ELSE IF S$=CHR$(&H1C) AND X1<502 T
HEN X1=X1+16
1220 IF S$=CHR$(&H1E) AND Y1>34 THEN Y1=Y1-8 ELSE IF S$=CHR$(&H1F) AND Y1<159 TH
EN Y1=Y1+8
1230 IF S$=CHR$(&H20) THEN IF U1>X1 AND U1<X1+56 AND V1>Y1 AND V1<Y1+20 THEN GOS
UB  1260 ELSE BEEP
1240 GOTO 1130
1250 RETURN 1040
1260 ´アタリ
1270 PLAY"T250V10S8M10000O6F4"
1280 PRINT@(16*(C+16),10),&H216A:C=C+1:LOCATE23,2:COLOR4:PRINT C
1290 IF C>HS THEN HS=C:LOCATE28,1:PRINT HS ELSE RETURN 1120
1300 IF HS<20 THEN RETURN 1120
1310 PRINT@(130,130),&H2561,&H2540,&H252B,&H2472,&H3C68,&H246A,&H2439,&H242E
1320 PRINT@(258,130),&H2446,&H244F,&H2424,&H2431,&H245E,&H243B,&H2473,&H2123
1330 RETURN 1250
```

漢字ROMがある場合のBASICチェックサム

```
LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
1000 00 15 81 7D 6D 72 00 EC :DE
1080 17 E7 E0 00 AE DD F4 33 :90
1160 20 51 CB FE E9 6A 1E B0 :5B
1240 FE A5 00 70 DB DD 9B F5 :5B
1320 E6 A8 00 00 00 00 00 00 :8E
---------------------------------
     1B 9A 2C EB DF 96 AD C4 /B2
```

BASICチェックサムについては本誌41ページを参照してください。

漢字ROMのない場合（変更行のみ）

```
1070 WIDTH80,25:COLOR4:CLS:SYMBOL(1,6),"メダ゛",2,2:SYMBOL(45,6),"カトリケ゛",2,2:SYMBOL
(116,6),"-ム",2,2
1140 LINE(U0,V0)-(U0+16,V0+16),PSET,0,BF:COLOR5:SYMBOL(U1,V1),"♠",2,2
```

```
1280 SYMBOL(16*(C+16),10),"♠",2,2::C=C+1:LOCATE23,2:COLOR4:PRINT C
1310 SYMBOL(130,130),"ﾒﾀﾞｶﾗ､ﾄﾘｽｷﾞﾃﾊｲｹﾏｾﾝ｡",2,2
```

1320行は不要

漢字ROMのない場合のBASICチェックサム

```
 LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
 1000 00 15 81 7D 6D 72 00 91 :83
 1080 F7 E7 E0 00 AE DD 59 33 :D5
 1160 20 51 CB FE E9 6A 1E B0 :5B
 1240 FE A5 00 70 7A DD 9B C4 :C9
 1330 A8 00 00 00 00 00 00 00 :A8
---------------------------------
      BD F2 2C EB 7E 96 12 38 /24
```

FM-8の場合の変更行

```
1270 FOR B=40 TO 10 STEP-10:BEEP1:FOR A=0 TO B:NEXT:BEEP0:FOR A=1 TO B:NEXT:NEXT
```

FM-11の場合の変更行

```
1 SCREEN 0
1270 PLAY"O6F4"
```

神経衰弱ゲーム ———— 秋元克紀(東京都江東区) for FM-7,8,11

トランプの神経衰弱です。RUNしたら，人数（2～20人），名前（8文字まで）を入力してください。カードの指定は，横座標，縦座標の順に2桁の数で行います。

ゲームが進行して，残りカードが8枚以下（2枚のときは除く）になると，カードが合うごとに自動的にカードをかきまぜます。こうしてゲームが終了すると各人の取ったカードの組数をグラフにして終わります。

編集部記：カードを並べるときに，乱数による箱入れ法でやっていますが，これでは時間がかかるので，トランプを切るように並べ変えのプログラムを使ったほうがよいと思います。FOR～NEXTループで，カウントする変数をループ内で操作するのは，プログラムの構造上，工夫したほうがいいですね。

神経衰弱ゲームリスト

©1984 K. AKIMOTO

```
1000 CLEAR 500:DIM T$(9,6),P(20),N$(20):RANDOMIZE TIME MOD 32767
1010 WIDTH40,25:CONSOLE 15,1,0,0:COLOR7,0:CLS:B$(1)="J":B$(2)="Q":B$(3)="K":R=27
1020 J$(1)="1st":J$(2)="2nd":INPUT "ﾆﾝｽﾞｳ";N:IF N<2 OR N>20 THEN 1010
1030 E(2)=2:E(3)=2:FOR I=1 TON:PRINT I"ﾊﾞﾝﾒﾉﾋﾄﾊ ";
1040 INPUT "ﾅﾏｴｦ ｲﾚﾃｸﾀﾞｻｲ";N$(I):N$(I)=N$(I)+"ｻﾝ ":NEXT:CLS:GOSUB 1210
1050 FOR P=1 TO N:WHILE W=0:FOR C=1 TO 2
1060 COLOR7:CLS1:PRINT N$(P);J$(C);:INPUT" CARD";I$
1070 X=VAL(LEFT$(I$,1)):Y=VAL(RIGHT$(I$,1)):IF X<1 OR X>9 OR Y<1 OR Y>6 THEN 106
0
1080 C$=T$(X,Y):IF C$="*" THEN 1060 ELSE GOSUB 1150:X(C)=X:Y(C)=Y:T$(X,Y)="*":NE
XT
1090 IF CA$(1)=CA$(2) THEN 1140 ELSE H$="ﾊｽﾞﾚ":H=5:COLOR4:T$(X(1),Y(1))=C$(1):T$
(X,Y)=C$
1100 BEEP:SYMBOL (392,151),H$,8,4,H:FOR A=1 TO 2000:NEXT A
1110 FOR I=1 TO 2:LOCATE X(I)*3+6,Y(I)*2+1:PRINT "■":NEXT
1120 CLS3:IF H=5 THEN NEXT:GOTO 1050 ELSE IF R>1 AND R<5 THEN GOSUB 1270
1130 IF R=0 THEN 1290 ELSE WEND
1140 P(P)=P(P)+1:R=R-1:FOR I=0TO50:BEEP1:H$="ｱﾀﾘ!":H=6:COLOR1:BEEP0:NEXT:GOTO 11
00
1150 COLOR 10:LOCATE X*3+6,Y*2+1:PRINT RIGHT$(STR$(C),1);:Z=C*178-125:C$(C)=C$
1160 LINE(Z,136)-(Z+120,199),PSET,7,BF:M=VAL(LEFT$(C$,2)):IF C$="" THEN 1200
1170 YB=4:N$=RIGHT$(C$,1):IF N$="0" THEN YB=2:N$="10"
1180 SYMBOL (11+Z,143),N$,YB,2,E(M):SYMBOL (37+Z,157),CHR$(&HE7+M),5,3,E(M)
1190 SYMBOL (75+Z,179),N$,YB,2,E(M):CA$(C)=N$:RETURN
1200 CA$(C)="j":SYMBOL (80+(C-1)*176,153),"J",8,4,3:RETURN
1210 FOR I=1TO4:FOR J=1TO13:A=INT(RND*9)+1:B=INT(RND*6)+1:IF T$(A,B)<>"" THEN J=
J-1:NEXT
1220 N$=STR$(J):IF J=1 THEN N$="A" ELSE IF J>10 THEN N$=B$(J-10)
1230 T$(A,B)=STR$(I)+N$:NEXT J,I
1240 CLS:COLOR 7:LOCATE 9,1:PRINT "1  2  3  4  5  6  7  8  9";
1250 FOR B=1 TO 6:COLOR7:LOCATE 5,B*2+1:PRINT B;:FOR A=1 TO 9
1260 COLOR (T$(A,B)="*")*3+4:PRINT " ■ ";:NEXT A,B:RETURN
1270 COLOR 7:PRINT"CARD ｦ ﾅﾗﾍﾞｶｴﾏｽ｡"+CHR$(7);:FOR Y=1 TO 6:FOR X=1 TO 9
1280 SWAP T$(X,Y),T$(INT(RND*9)+1,INT(RND*6)+1):NEXT X,Y:GOTO1240
1290 CONSOLE 0,25,0,0:BEEP1:SYMBOL (160,0),"GAME OVER!",4,2,4
1300 FOR P=1TON:BEEP1:LOCATE 0,P+2:COLOR 5:PRINT LEFT$(N$(P)+SPACE$(8),10);":";
1310 COLOR 3:PRINT STRING$(P(P),"□");:COLOR 7:PRINTP(P):BEEP0:NEXT
1320 LOCATE 22,N+4:PRINT "PLAY AGAIN <Y>":IF INPUT$(1)="Y" THEN RUN
```

LINE	+0	+1	+2	+3	+4	+5	+6	+7	SUM
1000	EF	DF	FA	56	7C	B4	01	BD	:0C
1080	7F	96	9D	B3	C7	A0	BA	A4	:2A
1160	39	D7	6C	F6	09	13	15	41	:E4
1240	39	9D	76	CE	4D	4A	01	9F	:51
1320	1E	00	00	00	00	00	00	00	:1E
	FE	E9	79	CD	99	B1	D1	41	/89

FM-11のための追加行

```
1 SCREEN 0
```

FMポーカーゲーム

川上和徳(所沢市)

for FM-7,8,11

これは1人で遊ぶポーカーゲームです。RUNするといきなりゲームが始まります。はじめの持ち金は，$1000です。まず，賭け金を聞いてきます。これに対して，0から9までの数字を入力してください。〝1″なら$100，〝2″なら$200というように賭け金が決まります。ただし〝0″は$1000になります。そのとき，たとえば持ち金が$100でも$1000まで自由に賭けられます。次にカードが配られ，どのカードを残すか聞いてきます。左から1番目のカードを残すときは〝1″，2番目のときは〝2″と入力します。間違えたときはスペースバーを押すことにより，もう1度入力し直せます。決まったらリターンキーを押すとカードが取り換えられ，同時に役の判定がされます。そして，配当金が持ち金に加算されます。持ち金が$0以下になるとゲームオーバーです。役の倍率は次のとおりです。

ワンペア	1倍
ツーペア	2倍
スリーカード	5倍
ストレート	10倍
フラッシュ	15倍
ロイヤルストレート	20倍
フルハウス	30倍
フォーカード	50倍
ストレートフラッシュ	100倍
ロイヤルストレートフラッシュ	500倍

なお，この倍率が気に入らないときは，プログラムリスト中のYの値を変えればいいでしょう。

編集部記：トランプゲームを作成するときに，たいへん役立つショートプログラムです。役の判定やカードの表示などもかなり工夫されています。カードの裏の模様もあったらよかったのにネ!!

FMポーカーゲームリスト

©1984 K. KAWAKAMI

```
1000 REM ═══ POKER Ver 2.6 ═══
1010 DEFINT A-Z:DEFSNG M,B:DIM C(100),T(5),E(5),CM$(5):RANDOMIZE TIME:WIDTH 40,2
5:CLS
1020 FOR Z=1 TO 52:C(Z)=Z:NEXT Z:M=1000:COLOR 4
1030 FOR Z=7 TO 2 STEP -5:SYMBOL(70+Z/5,2),"FM POKER GAME Ver 2.6",3,2,Z:NEXT Z
1040 FOR Z=1 TO 52:A=INT(RND(1)*52)+1:SWAP C(Z),C(A):NEXT Z
1050 FOR Z=40 TO 520 STEP 120:LINE(Z,28)-(Z+96,108),PSET,1,BF:NEXT Z
1060 B=0:GOSUB 1290:GOSUB 1300:LOCATE 8,24:PRINT"How much do you bet ?";
1070 A$=INKEY$:B=VAL(A$)*100:IF B=0 THEN IF A$="0" THEN B=1000 ELSE 1070
1080 M=M-B:GOSUB 1290:FOR Z=1 TO 5:GOSUB 1310:NEXT Z
1090 GOSUB 1300:LOCATE 7,24:PRINT"What cards do you hold ?";
1100 FOR Z=1 TO 5:T(Z)=0:NEXT Z
1110 A$=INKEY$:IF VAL(A$)>0 AND VAL(A$)<6 THEN Z=(VAL(A$)-1)*120+40:SYMBOL(Z,110
),"HOLD",3,1,3:T(VAL(A$))=1
1120 IF A$=" " THEN FOR Z=1 TO 5:T(Z)=0:NEXT Z:LINE(0,110)-(639,118),PRESET,0,BF
1130 IF A$<>CHR$(13) THEN 1110
1140 FOR Z=1 TO 5:IF T(Z)=0 THEN SWAP C(Z),C(Z+5):GOSUB 1310:NEXT Z ELSE NEXT Z
1150 FOR Z=1 TO 4:FOR X=Z+1 TO 5:IF E(Z)>E(X) THEN SWAP E(Z),E(X)
1160 NEXT X,Z:Y$="NO PAIR":Y=0
1170 IF CM$(1)=CM$(2) AND CM$(2)=CM$(3) AND CM$(3)=CM$(4) AND CM$(4)=CM$(5) THEN
 Y$="FLUSH":Y=15
1180 IF E(1)=E(4) OR E(2)=E(5) THEN Y$="FOUR CARDS":Y=50:GOTO 1250
1190 IF E(1)=E(3) AND E(4)=E(5) OR E(1)=E(2) AND E(3)=E(5)THEN Y$="FULL HOUSE":Y
=30:GOTO 1250
1200 IF E(1)=E(3) OR E(2)=E(4) OR E(3)=E(5) THEN Y$="THREE CARDS":Y=5:GOTO 1250
1210 IF E(1)=E(2) AND E(3)=E(4) OR E(1)=E(2) AND E(4)=E(5) OR E(2)=E(3) AND E(4)
=E(5) THEN Y$="TWO PAIRS":Y=2:GOTO 1250
1220 IF E(1)=E(2) OR E(2)=E(3) OR E(3)=E(4) OR E(4)=E(5) THEN Y$="ONE PAIR":Y=1:
GOTO 1250
1230 IF E(1)=E(2)-1 AND E(2)=E(3)-1 AND E(3)=E(4)-1 AND E(4)=E(5)-1 THEN IF Y=15
 THEN Y$="STRAIGHT FLUSH":Y=200:GOTO 1250 ELSE Y$="STRAIGHT":Y=10:GOTO 1250
1240 IF E(1)=1 AND E(2)=10 AND E(3)=11 AND E(4)=12 AND E(5)=13 THEN IF Y=15 THEN
 Y$="ROYAL STRAIGHT FLUSH":Y=500:GOTO 1250 ELSE Y=20:Y$="ROYAL STRAGHT":GOTO 125
0
1250 SYMBOL(50,124),Y$+" ･･･"+STR$(Y)+" ﾊﾞｲ",2,2,2:SYMBOL(50,144),"ﾊｲﾄｳ ･･･"+STR
$(B)+" x"+STR$(Y)+" = $"+STR$(B*Y),2,2,5
1260 M=M+B*Y:B=0:IF M=<0 THEN LINE(120,52)-(520,120),PSET,5,BF:SYMBOL(140,60),"G
AME OVER",5,6,2:SYMBOL(250,104),"Push any Key !",2,2,1:A$=INPUT$(1):RUN
1270 GOSUB 1300:LOCATE 8,24:PRINT"═══ PUSH SPACE KEY! ═══";
1280 A$=INKEY$:IF A$<>" " THEN 1280 ELSE LINE(0,25)-(639,165),PRESET,0,BF:GOTO10
40
1290 LINE(0,167)-(639,186),PSET,6,BF:SYMBOL(60,170),"Money･･$"+STR$(M)+"  Bet･･$
"+STR$(B),2,2,0:RETURN
1300 LOCATE 0,24:PRINTSPC(38);:RETURN
1310 CM$(Z)=MID$("♥♦♠♣",INT((C(Z)-1)/13)+1,1):IF CM$(Z)="♥" OR CM$(Z)="♦" THEN C
C=2 ELSE CC=0
```

```
1320 E(Z)=(C(Z)-1)MOD13+1:CN$=MID$("A234567890JQK",E(Z),1):IF CN$="0" THEN CN$="
10":X=3 ELSE X=6
1330 LINE(40+(Z-1)*120,28)-(136+(Z-1)*120,108),PSET,7,BF:SYMBOL(61+(Z-1)*120,36)
,CM$(Z),6,4,CC:SYMBOL(63+(Z-1)*120,74),CN$,X,4,CC:RETURN
```

LINE	+0	+1	+2	+3	+4	+5	+6	+7	SUM
1000	00	A9	BD	EA	37	11	96	BB	:E9
1080	DF	58	00	1B	16	16	A2	23	:43
1160	FA	5D	F0	40	6C	E3	01	71	:48
1240	55	73	18	78	9C	F0	38	97	:B3
1320	C4	89	00	00	00	00	00	00	:4D
	F2	5A	C5	BD	55	FA	71	E6	/74

FM-11のための追加行

```
1 SCREEN 0
```

タイプ練習プログラム——山田真也(名古屋市)　for FM-7,8,11

このプログラムは，ただRUNして文字を入力するだけ。キーイングの練習をするだけのものです。アルファベットのときはCAPキー，カナのときはカナキーを押してください。なお，MISS1は×の数，MISS2は入力ミスです。

編集部記：タイプ練習に，時間制限と誤ったキーを押したときの処理を加えて評価してあるので，短い割に実用的なプログラムだと思います。もう少し表示を工夫すれば，よりGOOD！です。

変数の使い方に少し無駄があるようです。ループのためや入力のための一時的な変数をもっとまとめたほうがいいと思います。また，プログラムに用いる変数は，必ず値を定義してから実行するほうが，無難でしょう。なぜならばRUNを実行すると，変数の初期値は0となるのですが，プログラムの考え方として，それまで定義されていなかった変数をIF文にかけたり，+1にしたりするのは避けたほうが賢明です。

プログラムの流れにも多少無理があるようで，GOTOのスパゲッティになっている部分があります。処理について，よく整理してからプログラムを組むように心がけるべきでしょう。でも14歳という山田君の年齢を考えれば，将来が楽しみです。

タイプ練習プログラムリスト

©1984 S. YAMADA

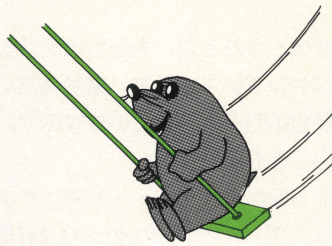
```
1000 WIDTH40,20:CLS:RANDOMIZETIME:DEFINTA-Z:DIMA$(20)
1010 INPUT "ALPHABET-0 OR ｶﾅ-1";K
1020 IFK=1 THEN C=44:B=177 ELSE C=26:B=65
1030 LOCATE0,1:INPUT "ﾚﾍﾞﾙ(1-10) ";L:LE=L*50
1040 IF L<1 OR L>10 THEN 1030
1050 FOR A=0 TO 19
1060 R=INT(RND(1)*C)+B
1070 A$(A)=CHR$(R)
1080 PRINT A$(A);" ";:NEXT:A=0
1090 I$=INKEY$
1100 IF A$(A)=I$ THENSC=SC+1:C=5:BEEP:GOTO1170
1110 IF TI>=LE THEN I$="X":C=2:MI1=MI1+1:GOTO1170
1120 IF I$="" THEN TI=TI+1:GOTO 1090
1130 IF I$<>A$(A) THEN MI2=MI2+1:TI=TI+1:C=2
1140 COLOR C:PRINT I$;:LOCATEX,3:C=7
1150 IF A=20 THEN GOTO 1180
1160 GOTO 1090
1170 X=X+2:A=A+1:TI=0:GOTO1140
1180 LOCATE5,6:COLOR7
1190 PRINT "POINT ";SC;"MISS1 ";MI1;"MISS2 ";MI2
1200 P=SC+MI2:S=SC/P*100:P1=SC/20*100
1210 LOCATE2,9:PRINTP"ｶｲ ﾆｭｳﾘｮｸ ｼﾃ ";SC;"ｶｲ ﾃｷﾁｭｳ", "ﾃｷﾁｭｳﾘﾂ";S;"%"
1220 LOCATE5,12:PRINTP1;"ﾃﾝ":END
```

LINE	+0	+1	+2	+3	+4	+5	+6	+7	SUM
1000	88	6E	EC	07	C7	A2	0B	D8	:35
1080	7B	4E	A5	8A	78	BE	CB	A1	:9A
1160	03	1C	3F	9E	8A	8F	B0	00	:C5
	06	D8	D0	2F	C9	EF	86	79	/94

ショートプログラム応募要領

- 1行80字以内で，できるだけ30行に収めてください。
- きれいなプログラムを望みます。マルチステートメントは多用しないでください。
- プログラムは，カセットテープまたはディスケットにいれてお送りください。ディスケットの場合は，新しいディスケットをお返し致します。
- プログラムの説明，使い方など必要なことを簡潔にまとめてください。
- プログラムおよび原稿は，多少修正させていただく場合があります。
- 他誌との二重投稿，盗作などモラルに反する応募は固くお断わり致します。
- 応募原稿は原則としてお返し致しません。
- 応募された方全員にOh！FMシールとシャープペンシルをお送り致します。
- すぐれた作品は本誌に掲載し，原稿料にかえてOh！FM特製Tシャツと図書券5,000円分をお送り致します。
- 原稿にも，住所・氏名・年齢・職業などを明記して，〒102東京都千代田区四番町2-1　㈱日本ソフトバンクOh！FMショートプログラム集係までお送りください。

READER'S AREA



▶月刊になるのがうれしい。(糸魚川市・山田信・17)

▶月刊になるなんて最高だぜい!(豊島区・森田尚幸・23)

▶Oh!FMの月刊化,喜んでます。(横浜市・笠井紀克・13)

――「月刊化おめでとう」のひと言で名前が載るなんて,恐らくこれが最初で最後ですヨ。次からはいっしょうけんめい感想を書いてネ! 多くの読者より月刊化お祝いの声をいただきました。これからも編集部一同,頑張っていきますので,今後ともご愛読のほどよろしくお願いします。

▶ディスクを所有しない者にとってVol.7は,つらい号でした。(目黒区・鈴木悦夫・27)

▶7号はディスクのない人には魅力がなかった。(奈良市・滝文雄・26)

――前号の「ディスク活用術」は,人によって反響はさまざまでしたが,この特集はみなさん熟読していただけたようですね。ディスクをまだ持っていない人にこそ,読んでいただきたいのです。そこんとこヨロシク!!

▶Oh!FMにプレゼントコーナーがあったなんて知らなかった。気づいていれば,もっと早く出したのに……。うーん,グヤジイー。(春日市・寺田祥治・16)

――そうなんです。FMは内容が濃いうえにプレゼントコーナーまであるのデス!

▶コンピュータによるミュージック作りに関する記事を載せてください。(京都市・山本純子・23)

――いいとも!! 6月号は特集が「音楽」の予定です。お楽しみに!

▶ついに次号から月刊化! うれしい♥これで〝虫つくろい〟が早く手に入る。素直なうれしい感想です。ルンルン!(名古屋市・小塚道代・20)

▶創刊号から愛読しています。内容も回を増すごとに充実してきて,うれしく思っています。これからもスタッフ一同,ますます頑張ってください。何といってもFMのことしか載っていないのがいいなあ。(町田市・田中宣彦・24)

――ありがとう。これからもぐわあんばりま~す。オーディオのFMファンが間違えて買っていくというウワサもあります。

▶去年,父にFM-7を買ってもらいました。なんといってもFM-7のそばにタモリがいっしょに写っているのが最高です。今ではFM-7を親友のように思っています。(湯沢市・藤岡芳樹・12)

――芳樹クン,いっしょうけんめい勉強してパソコン界の星になるのだよ!

▶暑中おみまい申しあげます。(羽島郡・伊藤基春)

――なっ,何を考えとるんじゃオヌシ! あげくのはてハガキのおしまいに「蓉子お姉様によろしく,ムフ♡」だって。キミの名前,よく覚えておこう。

▶わたしは廊下にマイコンを置いているので冬はとても寒いのです。長い時間,マイコンで遊んでいると過熱してくるので,つい私は「あ~,あったかい」とマイコンにひっついてしまうのです。それを見た弟が「ヘンタイ!」と言うのですが,私はそうなのでしょうか。(千葉市・遠藤美佐子・23)

――だいじょうぶです。私などは,寒い夜湯たんぽ代わりにFM-7を抱いて寝ています。

▶(真理ちゃんへ)私はFM-7を購入してからほとんどゲーム用に使用してきましたが,最近はゲームだけでは飽きてしまい,何かほかの使い方はないか考えています。よかったら,これからワープロ用に使用するのはどうですか! 私もワープロを始める予定です。(宇部市・藤本逸雄・31)

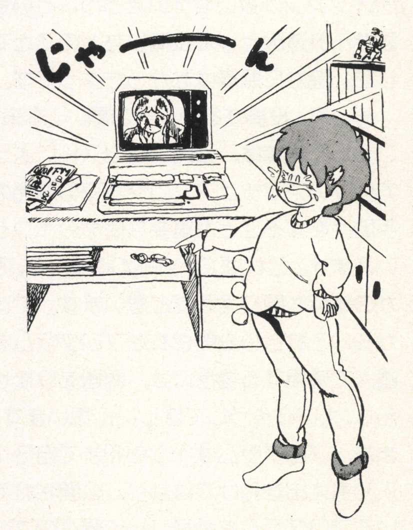
――ドキッ!(あわててVol.7のREADER'S AREAをめくる)―ありました。西宮市の三好真理さんへのご返事でした。(M)

▶いつも買っています。もう少し記事を増やしてください。とくに,製作記事です。毎号作るのが楽しみです。(高田郡・杉岡弘・17)

――今月号のオーディオタイマはどうでしたか? いろいろ作ればおもしろさ倍増。

▶いま,僕は機械語プログラムにムチュー。毎日,毎日なんで0~Fまでの数字とアルファベットで,「なんでこうなるんや?」とか「このプログラムを作った人は…?」なんて思っている。(神戸市・前川雄樹・13)

――A~Zの英語や0~9までの数字のほうも頑張ってよ!



ついにやったぜい!!
(熊本市・古川一成・16)

▶バグに関する質問があったので編集部に電話したら,なんと女性が出てきた。オレ頭にきて「女,子供に用はねえ!」って怒鳴ろうとしたら,ナッ,ナント彼女はオレの質問にすべて親切に答えてくれた。驚いたなあ!最近の女性の進歩はパソコンの進歩より早いんじゃないかなあ。後日,用はなかったけど電話したら,ずーっと話中で通じなかった。用もないのに長電話しないでくれ!(横浜市・安部毅・18)

――キミは感覚が遅れてるよー。今どきは

女だからどうだとか，男だからどうとかもうハヤンないよ。なお，電話はたくさんの読者の方々からきていますので，お話中のこともあります。

▶ Vol.7 125ページのショートプログラム「高速セーブ／ロードプログラム」(石川嘉信氏) の機械語部分はI／O '83年5月号の「FM-7ボーレートアップ用プログラム」(安田弘幸氏) の転用と思われるのでお知らせします。初心者が悪意なくしたものと思われますが，これからは注意してください。(入間市・草野義博)

——同じようなご指摘を多くの読者よりいただきました。調査の結果，ご指摘のとおりでした。安田氏とI／O編集部および読者にチェックが行き届かなかったことをおわびするとともに，ご注意をいただいた多くの読者に感謝いたします。さっそく，石川氏に問い合わせたところ，このようにすれば初心者を含めた多くのFMファンが扱いやすいだろう，との意図から改造版として投稿したとのことでした。確かに悪意はなかったようです。

そこで，投稿する方々へお願い。本来，こうした雑誌は，公開したリストによって個々のアマチュアプログラマの開発の手助けをすることが重要な使命の1つといえます。これはOh!FMに限らず，ほかの雑誌も同じ考えだと思います。ですから，これらの公開されたプログラムを個人で使用する場合には，問題ありませんし，大いにそうしてほしいと思います。また，プログラムは1から10まで自分1人で考え出すわけではなく，公開されているプログラムを参考にして個々のプログラミングの上達があるでしょうし，こうしてソフトの発展とパソコンファンの輪が生まれると思います。しかし，十分消化して自分のものとなったプログラムでも，投稿されるときには，ぜひ参考文献などを記載してください。これは先に発表した人に対しての礼儀だけではなく，ほかの読者が参考文献を参照することによって得られるものも大きいはずです。また，改造などの場合は，はっきりと，どの雑誌のどの号のどの記事の改造版と原稿に記してください。パソコンの発展のため，モラルだけはきちんと守りましょうネ。

▶ Vol.6の93ページ，タイトルペイントプログラムについて電話質問したところ，バグはないとの返答。だが，やはりあったのだ。このような質問の場合，門前払いにするのはやめてほしい。(大田区・平林雅英・23)

——失礼しました。確かに動いていたんです，ディスクで。テープでやってみるとなんとSyntax Error！ごめんなさい。

▶ Oh!FMがついに月刊化するそうで，おめでとう。今，北海道では横路知事をはじめ，500万全道民あげて喜んでいます。3月18日を国民の祝日にしましょう！(札幌市・勝又秀夫)

——それはいい考えです。となると当然，編集部も休めるわけですよネ。

▶ Vol.6の特集IIの中の「トッカータとフーガニ短調」はこれが投稿だけに，より注目できるものと思います。こうした実力のある人が増えるのを助けることのできる雑誌になることが必要ではないでしょうか。特集は色刷りになっていて読みやすく，たいへん充実しているようです。しかし，それ以外の記事が軽視されているのでは，と少々感じたりもします。特集とそれ以外の記事のほどよいバランス——これがよりいっそう全体のレベルをアップするのではないでしょうか。(東海市・杉山博昭・21)

——投稿にはかなりのレベルのものも少なくありません。今後ますます楽しみです。

▶もう，涙，涙。僕はパソコンをみるとよだれが出ます。もちろん，Oh!FMも感動です。(横浜市・小林章良・27)

▶かめばかむほど味がでる……。(豊橋市・山本景三・17)

——3度の食事をきちんととるようにしてください。

▶お金がもったいないから，今月こそ買うのやめよう…と思っているんですが，気がついたらいつも買っている。ところで，このハガキ，切手がいらないのがいいですね。読者を裏切って「切手をおはりください」なんてことにならないようにお願いします。(船橋市・石塚健彦・17)

——はがきの表を赤でフチどりする人がいます。色を着けると違反になってしまうのです。出版社を裏切るこういう人がいると……考えちゃうな！

▶話題が集中しているモグラの名前ですが，(堀りでェ)→holidayなんてのは，明るくていいんじゃないですか。(宮崎市・瀬戸啓充・15)

——おっ，いいじゃないか!! それではモグラの名前，「ほりでェ」にしよう！ たまには編集部も休みたいしね。瀬戸クンをはじめ，みなさんたくさんのアイデアどうもありがとうございマシタ。

EXPRESS

仲間募集

▶パソコン歴1年の35歳のFMユーザーです。情報交換，相談などできる友人を求めています。年齢などいっさい問いません，ご連絡ください。(〒112東京都文京区大塚3-42-8 菊島義雄)

▶1人でやっていないで，みんなで情報交換しましょう。往復ハガキでお知らせください。(〒526滋賀県長浜市北船町3-3 中島俊展)

▶「FMマイコンクラブ」を作りました。FMファンの方々，参加してみませんか。講習会を開いたり情報交換中心に活動。ビギナーも気軽に。(〒516三重県伊勢市楠部町字緑ケ丘3163 ☎0596-24-7734 林宏二)

▶大阪，堺でFM-7のユーザーが集まってクラブを作っています。会員30名。データの交換や最新ソフト情報などでたいへん充実しています。機関誌も発行しています。遠方の人も歓迎します。電話またはハガキで。(〒591大阪府堺市新金岡町2-6-4-206 ☎0722-51-3577 堺FMクラブ会長・阿部敏行)

売る

▶ FM-8用バブルカセットホルダ＋バブルカセット×2を5万円くらいで。(〒473愛知県豊田市花園町五反田6-1寺田コーポ103号 ☎0565-53-3669 奥村暢朗)

▶エプソン漢字プリンタMP-80K＋FMシリーズ用接続ケーブル＋付属品一式を8万円で。購入は58年1月。(〒981-14 宮城県角田市君萱字薬師堂32-2 ☎02246-8-2171 佐藤英夫)

買う

▶富士通のオリジナルアームロボを1万5000円で。(〒175東京都板橋区高島平4-32-8 魚住学司)

▶ FM-7用プリンタを3万円くらいで。漢字ROMも適価にて。まずは，ハガキを。(〒596大阪府岸和田市尾生町870-2 下村康夫)

▶ Oh!FM創刊号を2000円前後で。送料そちら持ち。往復ハガキで。(〒168東京都杉並区和泉2-27-19 伊藤隆司)

ほりでェ君のパッチング

Vol.7でお約束した「ディスクエディタZAPFM」はFM-8，FM-11用の変更個所と，「FLEX→F-BASICファイルコンバータ」のFM-8用変更個所です（リスト中の色部分）。

なお，FM-11用ZAPFMのリードコマンドの形式が一部変更になりました。

0サイド

R␣トラックナンバ␣セクタナンバ␣0⏎

1サイド

R␣トラックナンバ␣セクタナンバ␣1⏎

●ZAPFM……FM-11用

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
5128	20	E2	81	51	26	03	7E	59	:D4
5130	B2	81	57	26	05	17	01	51	:1E
5198	17	06	09	84	3F	81	31	27	:C2
51B0	00	E6	7D	58	A3	27	03	17	:9F
5590	55	DA	FD	55	D8	17	00	59	:C9
5798	12	12	34	67	BD	58	A4	35	:AD
5818	01	39	34	02	BD	59	72	35	:2D
5820	82	34	06	BD	59	59	35	86	:E6
5828	BD	59	7C	39	34	67	BD	59	:7C
5830	A6	35	E7	30	8D	00	05	17	:9B
5838	01	79	39	12	10	00	58	42	:6F
5840	00	0A	00	00	01	07	50	19	:7B
5848	00	19	00	01	7D	58	A3	26	:B8
5878	73	58	A3	20	25	7F	58	A3	:2D
58A0	0A	00	39	00	34	26	30	8D	:5A
58A8	00	37	31	8D	00	2D	C6	06	:EE
58B0	A6	A0	A7	80	5A	26	F9	30	:16
58B8	8D	00	18	4F	C6	06	ED	04	:B1
58C0	AD	9F	FB	FA	EC	04	30	8D	:EE
58C8	00	17	6F	8B	30	03	17	FF	:5A
58D0	5B	35	A6	12	00	58	E1	00	:81
58D8	00	00	78	1B	2E	40	0F	11	:21
58E0	07	00	00	00	00	00	00	00	:07
5958	00	34	02	8D	04	35	02	20	:1E
5960	04	44	44	44	44	84	0F	8B	:32
5968	30	81	39	23	02	8B	27	16	:D7
5970	FE	B6	1E	89	34	04	8D	E1	:01
5978	35	02	20	DD	34	14	30	9F	:4B
5980	FB	FA	5F	34	02	AD	03	35	:6F
5988	02	7D	58	A3	27	06	C6	02	:6F
5990	AD	03	25	02	35	94	30	8D	:5D
5998	00	0A	AD	9F	FB	FA	E6	01	:32
59A0	26	F4	20	EA	17	00	34	02	:71
59A8	86	0D	8D	D0	86	0A	8D	CC	:D9
59B0	35	82	39	AD	9F	FB	FA	30	:61
59B8	8D	00	05	AD	9F	FB	FA	39	:0C
59C0	10	00	59	C6	00	05	00	00	:34
59C8	0C	06	D7	00	00	00	00	00	:E9

●ZAPFM……FM-8用

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
5128	20	E2	81	51	26	03	7E	D8	:53
5130	B8	81	57	26	05	17	01	51	:24
5798	0F	BF	34	67	BD	DA	50	35	:85
5818	01	39	34	02	BD	AF	E3	35	:F4
5820	82	34	02	BD	AF	E9	35	82	:C4
5828	BD	D3	52	39	16	00	74	12	:B7
5830	12	12	12	86	50	97	C3	86	:EC
5848	CB	CC	35	F7	7D	05	AC	26	:17
58A0	0A	00	39	34	02	86	0D	17	:23
58A8	FF	7E	86	0A	17	FF	79	35	:D1
58B0	82	16	51	00	00	00	00	00	:E9

●FLEX→F-BASICファイルコンバータ……FM-8用

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
4180	00	03	BD	D3	52	1C	FE	39	:38
4188	BD	D3	52	86	0A	BD	D3	52	:54
41A0	86	20	BD	D3	52	5A	10	26	:18
4338	DA	50	35	67	30	01	39	A6	:D6
43B8	BF	BD	AF	E3	35	82	34	02	:FB
43C0	0F	BF	BD	AF	E9	35	82	0F	:E9
43C8	BF	BD	D3	52	39	0F	BF	34	:DC
43D0	67	BD	9F	97	35	E7	86	50	:4C
43E8	34	77	BD	CB	CC	35	F7	34	:5F

INFORMATION

FM-Xモニタ抽選会

2月22日，東京・虎の門のスカイラブにおいて，FM-Xのモニタ200名の抽選が，ユーザ代表の久保博彦君と中橋誠君立ち会いのもと，厳正に行われた。応募総数は1万数千通だった。

当選者は，北海道苫小牧市の坂井洋史さん，東京都八王子市の高山勝夫さん，兵庫県姫路市の石川聡さん，大分県佐伯市の泥谷恵美さんはじめ200名の方々です。おめでとう！

虫つくろいのページ

本誌Vol.7に，以下の誤りがありました。訂正し，おわびいたします。
なお，色文字部分が訂正事項です。

●Vol.7　P.34　FLEX→F-BASICファイルコンバータリスト

```
        削除
6150 TRAD=256+PEEK(PLDADR+42)*256+PEEK(PLDADR+43)
```

●Vol.7　P.45　カタログ表示プログラムリスト

```
165 ON ERROR GOTO370 (60行は削除)
```

同　310行

```
310 PRINT:PRINTE;CO$+"2CLUSTERS CRASHED!!":FOR I=1 TO E:PRINT"CLUSTER";EC(I);"TR
ACK:";RIGHT$("00"+HEX$(ET(I)),2);" SECTOR:";RIGHT$("00"+HEX$(ES(I)),2):NEXT I:PR
INT
```

●Vol.7　P.42　ステッププレート変更プログラムリストI

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
30A8	DE	BD	8F	4B	BD	9B	50	1C	:39
3148	08	8E	C9	53	A6	E4	85	80	:41

●Vol.4　P.31　セルフアセンブラ

申しわけありませんが，もう少しお待ちください。

FM-7(,8)による 6809機械語の書き方[後]

草津　酔床

18 新しい機械語

後半戦では，前回ちょっとだけ出てきたインデックスのXとYを中心に，取り扱う機械語の命令を少し増やして，**表18**の命令を使ってみます。機械語の勉強でいちばん難しいところは，前半戦で終わっています。読者は**表18**を見ただけで，だいたいの内容は見当がつきますね。

なお，表の中で（X）とか（Y）とかいうのは，インデックスによる指定番地の内容のことです。たとえばX＝12EFのときには，(X)は12EF番地のメモリの内容です。

──それでは，まずチェックサムのプログラムを書いてみましょう。

19 チェックサムとは

BASICのIFとかPRINTなどは意味がありますから，このつづりを間違えてOFとかPLINTと書いたときは，見直せばすぐにわかります。

ところが機械語は16進数の連続で，人間にとって意味をもっていませんから，読み合わせはたいへんです。おまけにプログラムを書き間違えたままRUNさせたとき，BASICならばSyntax　Errorなどと表示して作業をやめるだけですみますが，機械語は1字でも違っていると，マイコンがなんの警告もなしにいきなり暴走して，せっかく打ち込んだプログラムをみんな消してしまったり，人間がBREAKなどのキーを押しても，ぜんぜん応答してくれなくなったりします。したがって，機械語プログラムを打ち込んだときは，RUNさせる前に，それが絶対にノーエラーであることを確認しておく必要があります。

チェックサムとは，プログラムを発表する人が，たとえばメモリの内容をあらかじめ8個分ずつ合計して，その結果を合わせて発表しておき，読者が自分でキー入力したデータの合計と比較して，読者のキー入力が正しいかどうかをチェックしようとするものです。

この場合において，合計を計算するにはふつうはアキュムレータのAかBを使用します。AもBも16進数で2桁ですから，足し算を繰り返せばたちまち「桁あふれ」になってしまいますが，かまわずどんどん足し算を続けて，最後に残った答えをチェックサムとします。別の言い方をするならば，正しい足し算の答えを求めて，16進数で最下位の2桁だけを取り出してチェックサムするといってもよいでしょう。

たとえば**表19**のようなデータでは，正しい答えはFF+03ですから102になりますが，最下位の2桁を取って，チェックサムは02となります。

本誌は，チェックサムとしてメモリ8個分の合計を使っています(Vol.3を参照)が，雑誌によっては16個分としたり，番地を加えたり，あるいは縦の合計を合わせて使うなど，統一されていません。

すぐおわかりのとおり，機械語が正しく

表18　6809主要な命令表

1　データを入れる命令

Aに××を	86　××
Bに××を	C6　××
Xに○○××を	8E　○○　××
Yに○○××を	10　8E　○○　××

2　コピー命令(1)

○○××番地→A	B6　○○　××
A→○○××番地	B7　○○　××
○○××番地→B	F6　○○　××
B→○○××番地	F7　○○　××
A→B	1F　89
B→A	1F　98

3　コピー命令(2)

	通　常	オートインクリメント
(X)→A	A6　84	A6　80
A→(X)	A7　84	A7　80
(X)→B	E6　84	E6　80
B→(X)	E7　84	E7　80
(Y)→A	A6　A4	A6　A0
A→(Y)	A7　A4	A7　A0
(Y)→B	E6　A4	E6　A0
B→(Y)	E7　A4	E7　A0

4　コピー命令(3)

○○××番地とその次→X	BF　○○　××
X→○○××番地とその次	BF　○○　××
○○××番地とその次→Y	10　BE　○○　××
Y→○○××番地とその次	10　BF　○○　××

5　計算命令(AおよびB)

	A	B
××を加えよ	8B　××	CB　××
××を引け	80　××	C0　××
××と比較せよ	81　××	C1　××
(X)を加えよ	AB　84	EB　84
(X)を引け	A0　84	E0　84
(Y)を加えよ	AB　A4	EB　A4
(Y)を引け	A0　A4	E0　A4
1を加えよ	4C	5C
1を引け	4A	5A
○○××番地の内容を加えよ	BB　○○　××	FB　○○　××
○○××番地の内容を引け	B0　○○　××	F0　○○　××

6　計算命令(XおよびY)

	X	Y
××を加減	30　88　××	31　98　××
Aを加減	30　86	31　86
Bを加減	30　85	31　85

7　相対ジャンプ(ブランチ)命令

無条件	20　××
ゼロのとき	27　××
ゼロでないとき	26　××
キャリーのとき	25　××
キャリーでないとき	24　××

8　サブルーチン関係の命令

○○××番地のサブルーチンにジャンプせよ	BD　○○　××
元へ戻れ	39

表19 チェックサムの例

Add	+0	+1	+2	+3	+4	+5	+6	+7	Sum
7200	00	00	00	00	00	00	FF	03	:02
7208	FF	00	03	00	00	00	00	00	:02
7210	00	00	03	00	FF	00	00	00	:02
7218	03	00	00	00	00	00	FF	00	:02

入力できたかどうかを調べるのに，チェックサムは完全なものではありません。たとえば，ある番地の8をBと間違え，その隣りの番地のBを8と間違えたときは，チェックサムはぴったり合ってしまいますね。機械語を入力したときは，必ず慎重に読み合わせを行い，チェックサムは最後の仕上げとして使うようにするのが安全です。

20 概略設計

ふつうのチェックサム表示では，メモリ8個分の内容を横にズラリと並べ，その右側にそれらの合計を書きます（**表19**参照）。

しかし，チェックサムを使うときの状況を考えてみると，

(1) 読者は既にモニタのMコマンド(あるいは16進キー入力プログラム）を使って，機械語の書き込みを終わっている
(2) モニタのDコマンドを使って，一応読み合わせをやった
(3) 今回のプログラムを使って，雑誌などのリストを片手にチェックサムを調べる
(4) チェックサムが違っていれば，リストに印をつけながら，終わりまでチェックサムを続ける
(5) ひとつも違っていなければ作業終了
(6) 違っているメモリ8個分について，モニタのMコマンドなどを使って，読み合わせをしながら訂正する
(7) 訂正した部分について，(3)以下の作業を繰り返す

という作業内容になります。ですからここはちょっと手抜きをして，メモリ8個分の表示はやめて，単にチェックサムの数値だけを画面に書くことにしても，べつに不自由はないでしょう。

プログラムの形式は，いつものとおり機械語まじりBASICとします。そうすると，BASICと機械語との事務分担は，

(1) 作業開始番地をBASICから機械語の方に知らせてやる
(2) 機械語でメモリ8個分のチェックサムを計算し，BASICに答えを返す
(3) その答えをBASICで画面表示
(4) 次の8個分のメモリについて，(2)以下の作業を繰り返す

ということになりますね。

21 作業反復の定石

一定作業を何回か反復しなければならないとき，BASICならばFOR～NEXTをつかえばよいので簡単ですが，機械語ではふつうは**図4**のような形にします。

あらかじめ反復回数をカウンタに入れておけば，その回数だけ「作業」を反復して，終わったら次へ進んでくれるわけですね。似たような定石がほかにもありますが，**図4**の定石だけでもしっかり頭に入れておけば，非常に便利です。

22 詳細設計

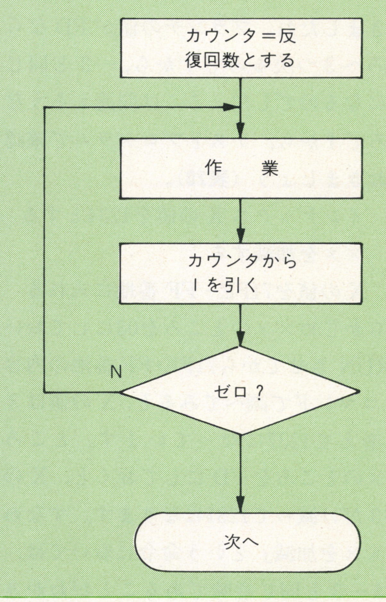
機械語側の作業内容としては，メモリ8個分の内容を合計して，その答えをBASICに返すのでしたね。

足し算のアキュムレータA，またメモリ8個分の合計を計算するにあたり，作業反復のカウンタはアキュムレータBを使うことにしましょう。また，BASICから作業開始番地を知らせてくるのですが，これは70BE・BF番地のメモリを使うことにします。たとえば70BE番地の内容が78，70BF番地の内容が00ならば，作業開始番地は7800番地というわけですね。

計算の答えをBASIC側に引き渡してやるときは，70BD番地を使いましょう。

そうすると，機械語側は次の作業をやることになります。

図4 作業を一定回数だけ反復



(1) 合計のAを0にしておく
(2) カウンタBを8にする
(3) 70BE・BF番地の内容をインデックスXに持ってくる
(4) Xの指定している番地の内容をAに加える
(5) Xに1を加えることにより，次の番地を指定させる
(6) カウンタBから1を引く
(7) 0でなければ(4)以下を繰り返す
(8) 0になれば，8個分のメモリの足し算が終わったのだから，答えを70BD番地に書き込む
(9) 次の作業のために，インデックスXの値を70BE・BF番地に戻す（Xは最初よりも8だけ大きくなっている）

23 符号付き加算

詳細設計の(5)で，Xに1を加える必要があります。これには**表18**の6を使うのですが，「××を加減」というとき，その××が0から7Fまでならば足し算で，80からFFまでならば引き算となります。前回の相対ジャンプのところで，似たような話が

表20　符号付き加算

```
200 CLEAR300,&H6FFF
210 DEF USR0=&H7100
220 V=USR0(V)
230 END
7100: 8E 12 34
7103: 30 88 XX
7106: BF 71 FE
7109: 39
```

出てきましたね。ブランチの値がFDならば後ろへ3つ戻るのですから，－3と同じ処理になるのでした。ここは間違えそうなところですから，テストプログラムで確認しておきましょう（**表20**）。

(1)　インデックスXの値を1234とする

(2)　××を加減する

(3)　Xの値を71FE・FF番地に入れる

あらかじめ××のところを03にしておいて，RUNさせてから71FE・FF番地の内容をDコマンドで調べてみると，Xの値は3だけ増えて1237になっています。ところが××のところをFDにしておくと，Xの値は3だけ減って1231になります。すなわち「××を加減」という命令においては，FDは－3と同じ意味であることがわかります。

このように，機械語の処理において，ときどき「7Fまでならば足し算，80以上ならば引き算」をしなければならないことがあります。機械語の本では，これを「符号付きの加算」とか「2の補数表現の加算」などと書いてあります。相対ジャンプの数値も符号付き加算ですね。

符号付きの加算では，7Fと80との間には大きな断絶がありますが，そのほかの部分では，数値の大きいほうが大きい数です。これもちょっと錯覚しそうなところですね。80は－80(16進)，FFは－1ですから，FFの方が大きいことになります。「符号付き加算において，0からFFまでのうち，いちばん小さい数はどれですか」と聞かれたとき，つい「FF！」と答えてしまいそうですが，それはまちがいです。

24 プログラムの完成

機械語プログラムは**表21**のとおりになりますが，これは先ほどの詳細設計に従って命令を並べていけばすぐできます。

これをBASICのDATA文に取り込んで全体のプログラムをまとめたのが**表22**です。チェックサムのプログラムが間違って

表21　チェックサム機械語

7080	BE	70BE	LDX	>$70BE	70BE、BF番地の内容をXに入れる
7083	86	00	LDA	#$00	A＝00とする
7085	C6	08	LDB	#$08	B＝08とする
7087	AB	84	ADDA	,X	Aに（X）を加える
7089	30	88 01	LEAX	<$01,X	Xに1を加える
708C	5A		DECB		Bから1を引く
708D	26	F8	BNE	L7087	0でなければジャンプ
708F	B7	70BD	STA	>$70BD	Aを70BD番地にストア
7092	BF	70BE	STX	>$70BE	Xを70BE，BF番地にストア
7095	39		RTS		リターン

表22　チェックサム

```
200 REM *** CHECK-SUM ***
210 CLEAR 300,&H6FFF
220 DEF USR0=&H7080
230 GOSUB 10000:´MACHINE LANGUAGE
240 INPUT "START ADDRESS";A$
250 A=VAL("&H"+A$)
260 V=INT(A/&H100)
270 POKE &H70BE,V
280 POKE &H70BF,A-V*&H100
400 PRINT
410 FOR I=1 TO 4
420 PRINT HEX$(A);": ";
430 FOR J=1 TO 8
440 V=USR0(V)
450 V=PEEK(&H70BD)
460 V$=HEX$(V)
470 IF LEN(V$)<2 THEN V$="0"+V$
480 PRINT V$;" ";
490 IF J=4 THEN PRINT "  ";
500 NEXT J
510 PRINT
520 A=A+&H40
530 IF I=4 THEN PRINT
540 NEXT I
600 PRINT "NEXT ADDRSS?(Y/N)"
610 GOSUB 9000:´YES/NO SCAN
620 IF E=0 THEN 400
800 END
9000 REM *** ´YES´ ´NO´ SCAN ***
9010 V$=INKEY$
9020 E=0
9030 IF V$="Y" THEN RETURN
9040 IF V$="y" THEN RETURN
9050 E=1
9060 IF V$="N" THEN RETURN
9070 IF V$="n" THEN RETURN
9080 GOTO 9010
10000 REM *** MACHINE LANGAGE ***
10010 A=&H7080
10020 FOR I=0 TO &H15
10030 READ D$
10040 D=VAL("&H"+D$)
10050 POKE A+I,D
10060 NEXT I
10070 RETURN
30000 REM *** MACHINE LANGUAGE DATA ***
30010 DATA BE,70,BE,86,  00,C6,08,AB
30020 DATA 84,30,88,01,  5A,26,F8,B7
30030 DATA 70,BD,BF,70,  BE,39
```

いたりすると，まるでマンガです。科学的な読者は，私の説明を頭から信用することなく手近な雑誌などのリストを見ながら，**表22**のプログラムが正しく動くことを確認してみなければなりません。

チェックサムのプログラムも，BASICだけで充分に実用になりますが，機械語を併用すると，苦労しただけのことはあって，スピードがだいぶ上がります。すなわちオールBASICでは，パラパラという感じで画面表示をするのに対し，本プログラムでは，バーッと一度に書いてしまいます。

25 コーヒーブレイク

良いプログラムとは，どんなプログラムでしょうか。参考書では，ふつうは次のようなものが挙げられています。

(1) 完全に動くこと
(2) 読みやすいこと
(3) 実行速度が速いこと
(4) プログラムが短いこと
(5) メモリの使用量が少ないこと（これは(4)にも関連するが，たとえば必要もないのに大きなDIM文を置いたりすると，プログラムは短くても，メモリの使用は多くなってしまう）

GOTO文だらけのスパゲッティ配線で，似たようなサブルーチンがあちこちに散在し，長くて読みにくく，おまけにバグまであるというのでは，悪いプログラムに決まっていますが，ある程度の水準に達したプログラムでは，必ずしも「あちらよりもこのほうが上」と決められないことがあります。たとえば読みやすさを考えてREM文を多く使うと，副作用として，実行速度は遅くなるし，プログラムは長くなり，またメモリの使用量も増えてしまいます。

またBASICのサブルーチンにおいて，最後に別のサブルーチンを呼んでからリターンするとき，

```
GOSUB ×××
RETURN
```

とする代わりに，単に，

```
GOTO ×××
```

とすると，メモリや実行速度の点で有利になる反面，読みやすさは犠牲になってしまいます。最初に挙げた良いプログラムの条件のうちで，一番大切なのは(1)の「完全に動くこと」です。そして，ほかの条件を注意深く見てみると，(2)の「読みやすいこと」というのは，(1)と密接な関連をもっていることがわかります。なぜなら，読みやすいプログラムならば，書いているときに頭の中がよく整理されているので，バグが入り込みにくいし，あとで万一バグを発見しても，その発生源を容易につきとめることができるからです。

こういうわけで，私たちがプログラムを書くときは，他の条件よりも，まず「読みやすいこと」を最優先に考えるべきだと思います。もちろん，メモリがどうしても足りないようなときは，なりふりかまわずプログラムを短くする必要があるかもしれませんが，機械語は使用メモリの少ないことが自慢なのですから，そういうピンチにぶつかることは，まずないはずです。

26 メモリ内容の引越し

最後に書くプログラムは，ある範囲のメモリに入っているデータを，そっくり他の番地に引越しさせようというものです。

たとえば，前回の16進キー入力プログラムを使って機械語をメモリに書き込んでいくときに，もしも1行とばしてしまったら，あとでそこから後の部分をずらして途中に割り込ませる必要があるし，逆にダブって打ちこんだならば，前へずらして詰めてやらなければなりません。また，自分で機械語プログラムを書くときでも，途中に割り込ませたり，削って間を詰めたりする作業が，しばしば必要になります。このため，市販の機械語開発用ソフトでは，メモリ内容の引越しの機能は，たいてい付いています。ただ名称はまちまちで，「エクスチェンジ」とか「トランスファ」とか，あるいは「ムーブ」などと呼ばれていますが，ここではいちおう「引越し」としておきます。

27 引越しの概略設計

引越しのやり方は簡単で，原データの先頭番地から内容を取ってきて，引越し先の先頭番地に移し，次に2番目から取ってきて引越し先の2番目に移し，以下同様にやって，原データの最終番地まで引越しが終わったら，作業終了です。

もっとも「以下同様」などといってもマイコンには通じませんから，厳密にいうならば，原データの先頭番地をP，最終番地をQ，引越し先の先頭番地をRとするとき，

(1) Pのところから内容を取ってきて，Rのところに移す
(2) P＝Qならば作業終了（1度で終わってしまったとすれば，原データはメモリ1つ分しかなかったことになる）
(3) PとRに，それぞれ1を加える
(4) (1)に戻る

という作業になります。

BASIC側と機械語側との事務分担ですが，P，Q，Rの値さえわかれば，あとの作業は機械語側だけでみんなできますね。

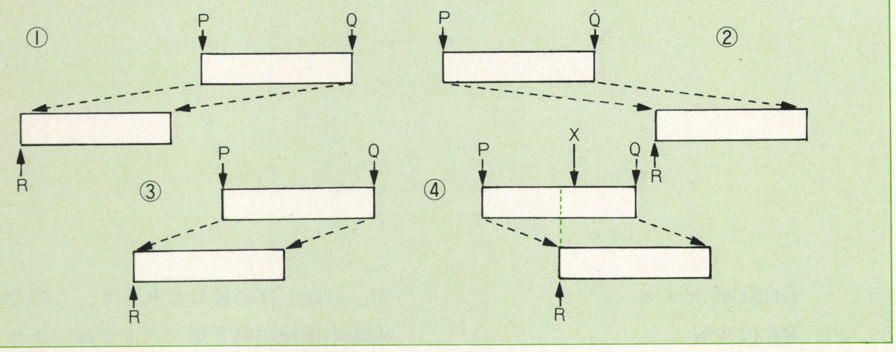
28 引越し禁止の場合

今の説明でわかるとおり，P<QでもP＝Qでもよいのですが，人間が間違えて終点よりも先頭のほうを大きくすると，P>Qになってしまって変なことになりそうです。

もうひとつ，大きな問題があります。

図5を見てください。原データと引越し先との相対的な位置関係は，①から④まで

図5　4種類の引越し



4種類がありますが，このうち④だけは，先ほどの方式で引越しさせることはできません。その理由はわかりますね。Pの内容を取ってきて，Rのところに移すのですがそこにはまだ別の原データが入っているのですから，結局×印以降のデータは，正規の引越しの前に，みんな壊されてしまいます。

どうしても④の引越しをやりたいならば，末尾のほうから移していくのですが，プログラムはだいぶ面倒になりますから，④の場合(すなわちP<R≦Qの場合)だけは，今回はあきらめることにしましょう。あきらめるといっても，いちおう少し離れた番地に引越しさせておいて，もう一度もってくればよいのですから，それほど困りません。

29 詳細設計

概略設計をかなり詳しくやりましたから詳細設計は楽ですね。

BASIC側から引き渡されるP，Q，Rとしては，原データの先頭番地Pは70FA・FB番地に入れてあり，最終番地Qは70FC・FD番地，引越し先の先頭番地Rは70FE・FF番地に入れてあるものとします。

インデックスXはいつも原データを，Yは引越し先を指すものとします。アキュムレータBは，今回は出番がありません。

(1)　70FA・FB番地から原データの先頭番地を取ってきて，Xに入れる

(2)　70FE・FF番地から引越し先の先頭番地を取ってきて，Yに入れる

(3)　Xによる指定番地の内容を，アキュムレータAにコピー

(4)　AをYによる指定番地にコピー。これでメモリ1つ分の引越しが完了

(5)　XとYに，それぞれ1を加える

(6)　70FA・FB番地の内容と，70FC・FD番地の内容を比較し，ちょうど同じならば，P=Qのところの引越しが終わったのだから，作業終了

(7)　作業終了でなければ，この次に(6)の比較をするときのために，Pの値を更新する。Pの値はXに入っていて，前よりも1だけ増えているので，これを70FA・FB番地に入れればよい

(8)　(3)に戻る

30 オートインクリメント

詳細設計の(5)で，X・Yに1を加える作業があります。これはチェックサムのときにもやったことで，「××を加減」という符号付き加算を使うのでしたね。

もちろんあのやり方で満点なのですが，**表18**の3にあるとおり，CPUの6809には「オートインクリメント」という便利な命令があるので，今回はこれを使ってみましょう。「オートインクリメント」とは，次のような意味をもつ命令です。

(1)　ふつうの命令と同じように，命ぜられた作業をせよ

(2)　作業を終えたら，インデックス(XまたはY)の値に1を加えておけ

つまり詳細設計の(3)と(4)のところでオートインクリメント付きの命令を使えば，(5)の命令は，わざわざ書かなくてもよいということになります。一般にインデックスXとYは，メモリの番地を指定するのに使われますが，その指定の仕方は，同一の番地をずっと指定し続けるのではなくて，たいていの場合は，1つずつ番地をずらしながら，次々に1つ大きい番地を指定していきます。こういう場合にオートインクリメント付きの命令を使うと，プログラムがだいぶ簡単で短くなるというわけですね。

31 機械語プログラム

それでは詳細設計にしたがって，機械語のプログラムを完成させましょう。**表23**のようになりますね。

詳細設計の(6)でP=Qかどうか調べるところだけは，70E0番地からの下請けサブルーチンとして切り離してあります。この下請けサブルーチンから戻ったときに，結果が0であれば70FA・FB番地の内容と70FC・FD番地の内容とが，完全に同じであるということですね。

コーヒーブレイクのときの話にも関連しますが，あるサブルーチンの中で，まとまった下請け作業があるときは，その部分を下請けサブルーチンとして切り離すほうが，プログラムは一般に読みやすくなります。もちろん「BD……(サブルーチンにジャンプせよ)」とか「39(元へ戻れ)」などの分だけメモリを余計に使ってしまいますが，なるべく読みやすさを優先させて書くことにしましょう。

32 プログラムの完成

BASICとしては**表24**のようになりますが，いちばん面倒なP，Q，Rの関係について，もう一度復習しておきます。

まずPとQの関係ですが，P<Qならば問題はありません。P=Qのときは，たった1つだけのメモリの引越しですから，このプログラムを使う意味はありませんが，使って悪いわけでもありません。P>Qのときはエラーです。

PとRについては，P>Rならば合格です。P=Rのときは，原データを全く同じ所に引越しさせるのですから無意味ですが，これも使って悪いわけではありません。P<Rのときは，次のQとRの検査が必要です。

QとRについては，Q<Rならば合格で，

表23 引越しの機械語

```
 70C0 BE   70FA  (70FA, FB番地)→X
 70C3 10BE 70FE  (70FE, FF番地)→Y
→70C7 A6   80    (X)→A  } オートインクリメント
 70C9 A7   A0    A→(Y)  }
 70CB BD   70E0  70E0番地のサブルーチン
←70CE 26   01    0でなければジャンプ
 70D0 39         リターン
→70D1 BF   70FA  X→(70FA, FB番地)
←70D4 20   F1    ジャンプ
 70E0
 70E0 B6   70FB  (70FB番地)→A
 70E3 B0   70FD  (70FD番地)を引く
←70E6 26   06    0でなければリターンへ
 70E8 B6   70FA  (70FA番地)→A
 70EB B0   70FC  (70FC番地)を引く
→70EE 39         リータン
```

P<R<Qのときは不合格です。P<R=Qならばどうでしょうか。Pのところのメモリの内容を取ってきて，Rのところに引越しさせるとき，Qの原データを変更してしまいますから，これも不合格です。

それでは本プログラムをRUNさせて，合格すれすれのP, Q, Rでテストしてみましょう。正しい処理をしていますか。こういうボーダーラインのところは，非常にバグが入りやすいので，とくに気をつける必要があります。このくらいのプログラムになると，BASICのPEEK／POKEではだいぶ無理で，少し長いデータを引越しさせようとすると，何十秒もかかってしまいます。ところが，ほんのちょっと機械語を使うだけで，ほとんどノータイムで作業が終わってしまうのですから，機械語は本当にすばらしいですね。

33 今後の勉強

今までの勉強で，基礎のところを固めたので，このへんで鶴岡哲明氏の解説（Vol. 1，2）などを読み直してみると，前よりはずっとよく理解できるはずです。

6809の命令としては，この連載で説明しなかったものも多いので，これからもっと機械語をやってみたいという方は，「6809マイクロプロセッサ・プログラミング・マニュアル（CQ出版社）」を買っておくとよいでしょう。このマニュアルは入門書と違って，読んですらすらわかるようなものではありません。いろいろな命令の具体的な動作内容がよくわからないときは,いままでいつもやってきたように，簡単なテストプログラムをつくって動作を確認し，その命令の性格を完全に理解してから,実際に使ってみるのがよいとおもいます。

しかし読者があわてて多くの命令を頭の中に詰め込もうとすることは，私はあまりおすすめしません。

表18の命令は，6809の命令の中で，もっとも中核的なものであって，これだけ知っていれば，たいていのプログラムは書けます。このほかにも便利な命令はありますが，その多くは，先ほどのオートインクリメントのように,「知らなくてもプログラムは書けるけれども，知っていればなおよい」という性格のものです。

BASICでも同じで命令をみんな暗記したからといってそれでプログラムがうまく書けるわけではありません。読者はまず**表18**の命令を自由に使いこなせるように，自分で具体的なプログラムを書きながら練習し，その間に，必要に応じて，少しずつ新しい命令を自分のものにするよう，守備範囲を広げていくべきでしょう。

なお私の説明では，無理に2進数の話を避けて通ってしまいました。ここから先に進むには，「2進数とは何か」というだいたいの知識が必要になりますが，これも少しずつ勉強して慣れてください。

表24 メモリ内容の引越し

```
200 REM *** EXCHANGE ***
210 CLEAR 300,&H6FFF
220 DEF USR0=&H70C0
230 GOSUB 10000:'MACHINE LANGUAGE
240 GOSUB 2000:'INPUT P,Q,R
250 GOSUB 2200:'POKE
400 V=USR0(V)
600 END
2000 REM *** INPUT P,Q,R ***
2010 PRINT "CURRENT DATA"
2020 INPUT "   FROM";P$
2030 P=VAL("&H"+P$)
2040 INPUT "   TO   ";Q$
2050 Q=VAL("&H"+Q$)
2060 IF P>Q THEN 2010
2070 INPUT "MOVE TO";R$
2080 R=VAL("&H"+R$)
2090 IF P>=R THEN RETURN
2100 IF Q<R THEN RETURN
2110 GOTO 2010
2200 REM *** POKE P,Q,R ***
2210 V=INT(P/&H100)
2220 POKE &H70FA,V
2230 POKE &H70FB,P-V*&H100
2240 V=INT(Q/&H100)
2250 POKE &H70FC,V
2260 POKE &H70FD,Q-V*&H100
2270 V=INT(R/&H100)
2280 POKE &H70FE,V
2290 POKE &H70FF,R-V*&H100
2300 RETURN
10000 REM *** MACHINE LANGUAGE ***
10010 A=&H70C0
10020 FOR I=0 TO &H2E
10030 READ D$
10040 D=VAL("&H"+D$)
10050 POKE A+I,D
10060 NEXT I
10070 RETURN
30000 REM *** MACHINE LANGUAGE DATA ***
30010 DATA BE,70,FA,10,  BE,70,FE,A6
30020 DATA 80,A7,A0,BD,  70,E0,26,01
30030 DATA 39,BF,70,FA,  20,F1,00,00
30040 DATA 00,00,00,00,  00,00,00,00
30050 DATA B6,70,FB,B0,  70,FD,26,06
30060 DATA B6,70,FA,B0,  70,FC,39
```

©1984 Y. KUSATSU

マシン語ゲームの作り方［II］

笠作貴弥

先号では，いきなりサブシステムがどうのこうのと出てきて少し戸惑ってしまった人もいると思うので，ここで，どうしてサブシステムを使わなければならないか考えてみます。

なぜサブシステムか？

FM-7では，先月号でも書いたように，メインCPUの負担を軽減するために画面制御やキー入力処理をメインCPUとは違った別の専用CPUが行っています。これをサブCPUと呼び，VRAM等を含めたシステム全体をサブシステムと呼んでいます。

サブCPUは，サブシステムモニタと呼ばれるプログラムを常に実行していて，メインCPUから共有RAMを通じてコマンドが与えられると，それに従って，画面に線を引いたり，文字を表示したりします。

このようなコマンドは，全部で三十数種用意されていますが，それぞれのコマンドは，BASICインタプリタ用に設計されているので，汎用性は高いのですが，高速ゲームのような特殊な用途には向いていません。ですから，高速のプログラムを作る場合は，それぞれのゲームに合った独自の画面制御プログラムをメイン側からサブシステムへ転送して，画面処理をしなければなりません。

サブシステムへプログラム転送・実行

では，具体的にサブシステムプログラムを転送し，実行する方法を書いてみたいと思います。これはOh! FM第3号にも載っていますが，今回のプログラムは，3号とは違う方法を用いています。

プログラムを転送するには，まず前項で触れた，サブシステムのコマンドの中から，メンテナンスコマンドというのを使います。このコマンドは，**図1**のようなデータを共有RAMにセットすることによって実行されます。**図1**においてのキーワードとは，FM-8，FM-11では，YAMAUCHIで，これが正しく合っていないと実行されません。FM-7では，8バイトのデータであれば，何でも構いません。また，メンテナンスサブコマンドとは，**図2**のような3種類があって，データやプログラムを共有RAM上から任意の場所へ転送したり，実行するというようなメンテナンスコマンドの仕事を決定するパラメータのことです。

では，具体的なプログラムの説明に入ります（**アセンブラソースリスト行番号419〜440**）。

①最初にサブシステムコマンドを送るためにサブシステムをHALTします（**442〜**）。

②次に共有RAM上の$FC80〜$FC8Eへ先に説明したメンテナンスコマンドをセットし(**463〜467**)，さらに，プログラム転送用のプログラム（**469〜480**）を$FC8F〜$FCACにセットします。

③次にサブシステムのHALTを解除します。これによって，サブCPUがメンテナンスコマンドを実行し，$D38F からのプログラム転送用のプログラムを実行します。

このプログラムは，一定のアドレス（$D3FE）を読んでいて，その内容が2であれば $D3FF のデータを取り出して，$C000 から順番にストアしていき，また，1であれば $C000 へ実行を移します。ですから，このとき，メイン側では，サブシステムをHALTして，$FCFEに2，$FCFFに転送するプログラムを1バイトずつセットして，HALTを解除するということを転送するバイト数回だけ繰り返して，最後に$FCFEに1をセットすればよいことになります。

これで，サブCPUは，サブモニタの制御を離れ，今，転送されたプログラムに実行が移ります（**図3**，**図4**参照）。

注）　　　　で囲まれたアドレスはサブ側のアドレスです。

サブプログラム

では，サブシステムを直接コントロールするプログラムを作ってみましょう。

この部分は，ハードウェアと密接な関係があるので，先月号や，マニュアルのシステム解説も一緒に参考にしてください。

ここは，最初の項でも述べたとおり，作るゲームによって独自のものを組まなければなりません。ですから，一般的に次の点に注意して，設計してください。

①プログラムサイズは，データを含めて

図1　メンテナンスコマンド

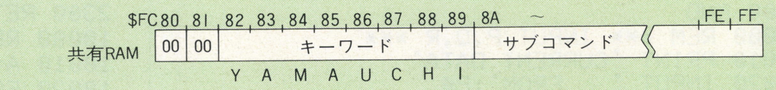
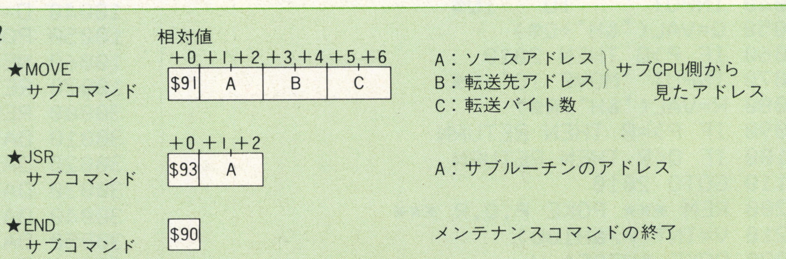


図2



4Kバイト以内。ただし，注意してプログラムを組めば，$D000〜$D37F も使用できます(割り込みで使用するワークエリアがある)。また，パレットを操作して，VRAMを1枚殺せば，さらに多くのエリアを使えます。

② DPレジスタの値は，変更しない。

(NMI)などの割り込み処理で，ダイレクトページアドレッシングを使っているので，変更すると暴走する可能性があります。

③リローケータブルに組む

アセンブルするときにアドレス計算が複雑になるので，リローケータブルに組んだ方が良いと思います。

では，具体例として本プログラム中のサブシステム・コントロールプログラムを説明します（482〜657）。

本プログラムでは，次の5つの仕事をサブシステムが受け持っています。

①画面の消去（568〜592）
②キー入力　（655〜657）
③文字列表示（523〜533）
④エイリアン表示（597〜653）
⑤キーデータ参照（521）

以上の処理をコマンドハンドラ（485〜503）が，共有RAMのコマンドレジスタ $D382 を読み取って各処理ルーチンをコールします。

○コマンドハンドラ

コマンドハンドラの働きを詳しく説明します（図6参照)。

①コマンドレジスタをクリアする（＝メイン側からのコマンド要求がない状態)。

②サブ側でコマンド受け付け可能なことをメイン側に知らせるためにREADY信号を送る（$D40A をリード)。

③コマンドレジスタをサーチして，0以外になるまで，サーチし続ける。

図3　フローチャート

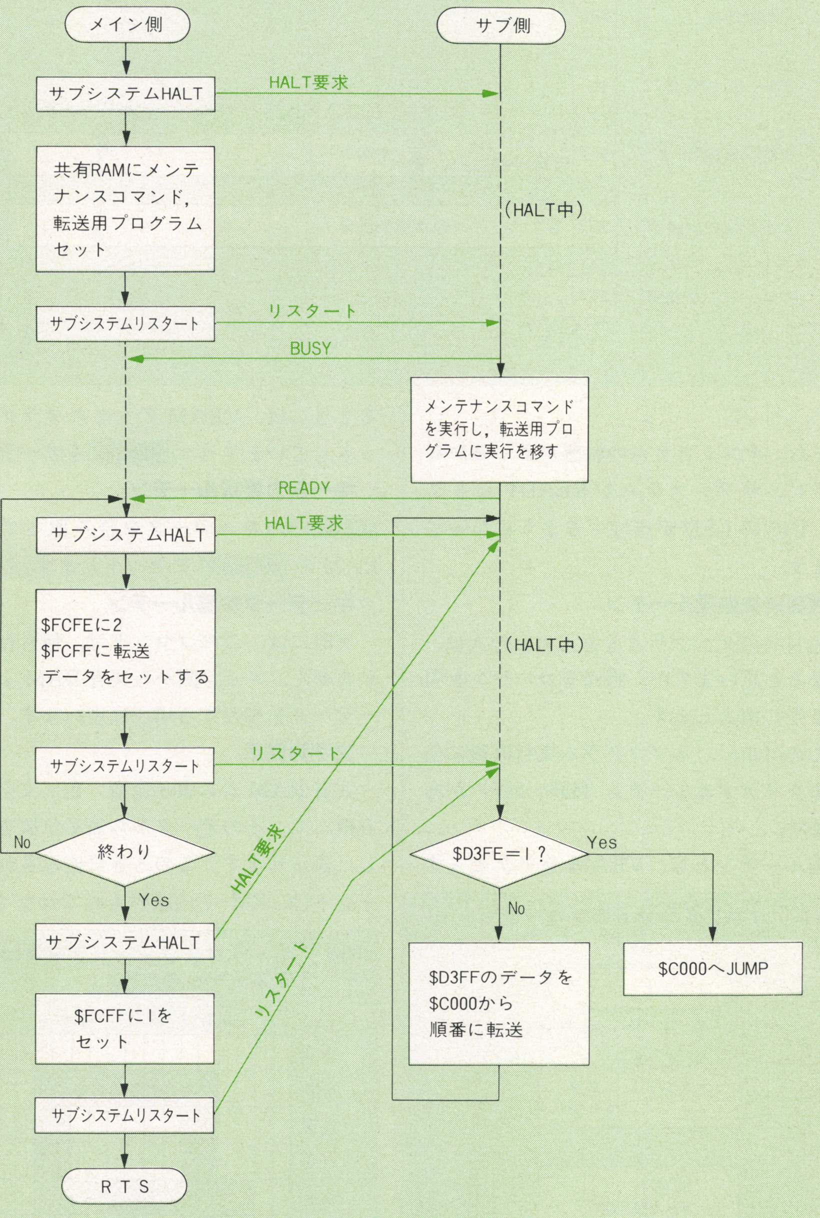
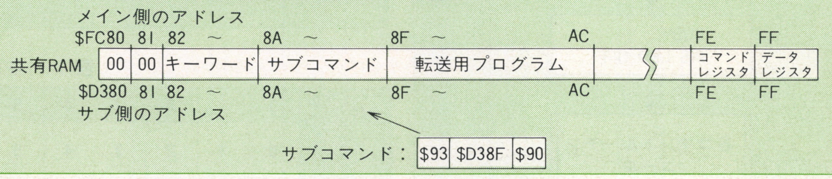


図4　サブプログラム転送時の共有RAMの様子

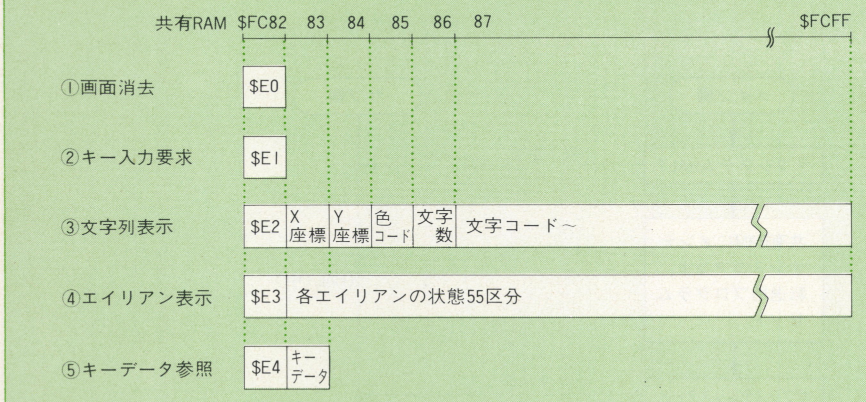


④コマンド要求があったら，次のコマンドを受け付けないように，BUSY信号を送る（$D40A をライトする)。

⑤読み取ったコマンドから各処理ルーチンをコールする。

⑥各処理をし終えたら，再び①から繰り返

図5 メイン⇨サブへのコマンドフォーマット



す。

また，サブシステムの状態がよく分かるように，サブシステムがREADYなときに，INSのLEDが点灯するようになっています。

○**画面消去処理ルーチン**

先月号のダンプリストを入力した人は，わかると思いますが，画面をカーテンを引くように消去します。

このほかに，本プログラム実行直後に全面をクリアするルーチン（**556～566**）があります。

両ルーチンとも，VRAMをアクセスするときには，VRAMアクセスフラグをセットしています（＝ $D409 をリード)。

○**キー入力要求ルーチン**

$D401 のキーデータを読み取って共有RAMの $D383 にセットします。

○**キーデータ参照ルーチン**

実際には，このコマンドは，何も行っていません。メイン側から共有RAM上のキーデータを参照する場合に用います。

○**文字列表示**

共有RAMから表示位置，色，文字を読み取って，そのデータから表示位置を計算し，表示します。このときに表示されるフォントは，FM-7内蔵のものではなく，新たに定義したものです(**図7，表1**参照)。なお，キャラクタフォントデータは，$5663～$5962となります（**図8**参照)。

○**エイリアン表示ルーチン**

メイン側から共有RAMに55匹のエイリアンの状態がセットされるので，これをCRT上へ投影します。ただし処理速度の関係から1回のコマンド要求に対して11匹のエイリアンを表示します。つまり5回このコマンドが要求されると，全部のエイリアンが表示されます。

以上でサブ側の説明は，終わります。

先月号で，I/O領域のVRAMオフセットレジスタ $D40E，0F の説明が不充分だったので，補足します。

このレジスタは，VRAMにオフセット

図6 コマンドハンドラフローチャート

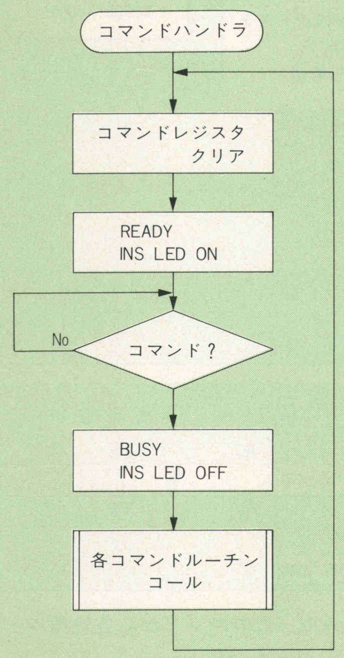
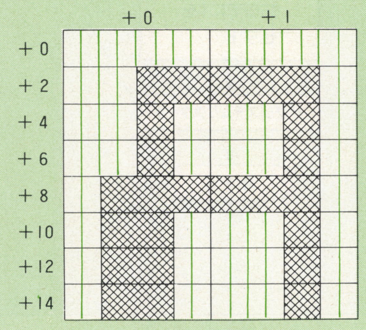


図7 キャラクタフォントデータのフォーマット



16バイトで1文字を構成

図8 サブプログラムメモリマップ（アドレスは転送前）

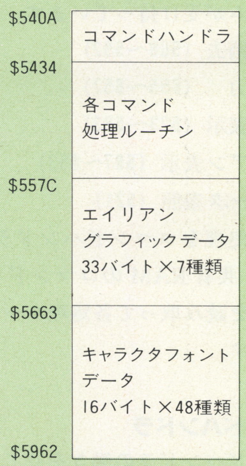
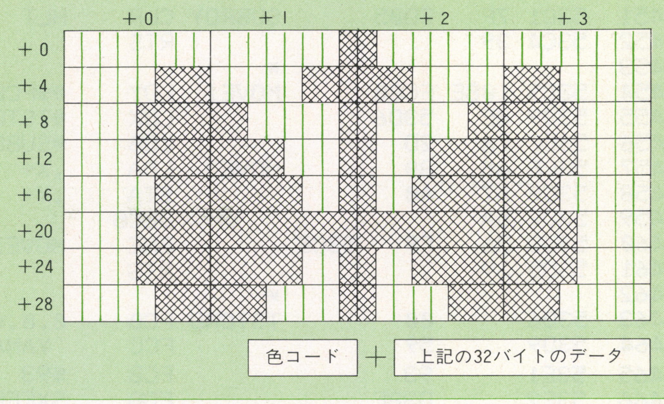


表1 キャラクタコード対応表

上位＼下位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
2	W	X	Y	Z	SP	●●	●	—	壁1	壁2	壁3	壁4	←	→	ボール	ラケット

図9　エイリアングラフィックデータのフォーマット



をかけ，ハード的にスクロールするためのもので，ソフトでスクロールするより非常に高速です。FM-7，8において，全画面のスクロールと画面の一部のスクロール速度が違うのは，このためです。

具体的にこのレジスタを利用するには，以前にセットした値に160(10)を足したり，引いたりすることで，2ドット単位で上下にスクロールすることができます。このレジスタはリードするとその値が不定となるので，必ず，ワークエリア上にセーブしておく必要があります。このワークエリアは，サブCPUがサブモニタの管理下にあるときは，FM-7で$D01F，20，FM-8では$D01D，1Eとなります。

©1983　T. KASASAKU

```
                              NAM    ALIEN
                        *
                        * ALIEN BREAK DOWN
                        *  Copyright (C) 1983
                        *             By ETF Laboratory.
                        *  PROGRAM Designed
                        *             By T.Kasasaku.
                        *
               5000     BEGIN  EQU   $5000
     5000                      ORG   BEGIN
                               OPT   NOG,SYMBOL
                        *
               FD05     HLT    EQU   $FD05
               1800     EWORK  EQU   $1800
               1837     VRAM   EQU   EWORK+55
                        *
     537F 1A   50       SUBMOV ORCC  #$50     Inhibit interupt
     5381 8D   2B  53AE        BSR   HALT     ┐TESTコマンドを使って
     5383 8D   40  53C5        BSR   MOVE1    ┘プログラム転送用プログラムを転送
     5385 8D   3A  53C1        BSR   SUBRDY
     5387 8E   5403            LDX   #SUBPRG        サブ側に転送するプログラムの先頭アドレス
     538A 108E 0560            LDY   #SUBEND-SUBPRG    〃    バイト数
     538E 8D   1E  53AE SMOVE1 BSR   HALT     サブCPUをHALT
     5390 86   02              LDA   #2       ┐
     5392 97   FE              STA   $FCFE    ┘データ転送を示すフラグセット
     5394 A6   80              LDA   ,X+      ┐
     5396 97   FF              STA   $FCFF    ┘データを共有RAMに書く
     5398 8D   27  53C1        BSR   SUBRDY   サブシステムリスタート
     539A 31   3F              LEAY  -1,Y     転送完了？
     539C 26   F0  538E        BNE   SMOVE1   No
     539E 8D   0E  53AE        BSR   HALT     サブシステムをHALT
     53A0 86   01              LDA   #1       ┐
     53A2 97   FE              STA   $FCFE    ┘転送完了を示すフラグセット
     53A4 8D   1B  53C1        BSR   SUBRDY   サブシステムリスタート
     53A6 8E   0000     WAIT   LDX   #0       ┐
     53A9 30   1F       WAIT1  LEAX  -1,X     │サブシステムの状態が整うまでWAIT
     53AB 26   FC  53A9        BNE   WAIT1    │
     53AD 39                   RTS            ┘
                        *
     53AE 34   02       HALT   PSHS  A
     53B0 B6   FD05     HALT2  LDA   HLT      サブCPU READY？
     53B3 2B   FB  53B0        BMI   HALT2    No
     53B5 86   80              LDA   #$80     ┐
     53B7 B7   FD05            STA   HLT      ┘サブCPU HALT
     53BA B6   FD05     HALT3  LDA   HLT      サブCPU HALT？
```

```
53BD 2A   FB    53BA          BPL   HALT3      No
53BF 35   82                  PULS  A,PC
                        *
53C1 7F   FD05          SUBRDY CLR  HLT        サブCPU HALT解除
53C4 39                       RTS
                        *
53C5 108E FC80          MOVE1 LDY   #$FC80
53C9 8E   53D6                LDX   #TSTCMD
53CC C6   2D                  LDB   #SUBPRG-TSTCMD
53CE A6   80            MOVE2 LDA   ,X+
53D0 A7   A0                  STA   ,Y+
53D2 5A                       DECB
53D3 26   F9    53CE          BNE   MOVE2
53D5 39                       RTS
                        *
53D6      00            TSTCMD FCB  0,0,$3F
53D9      59                  FCC   ´YAMAUCHI´
53E1      93                  FCB   $93
53E2      D38F                FDB   SXFER-TSTCMD+$D380
53E4      90                  FCB   $90
                        *
53E5 8E   C000          SXFER LDX   #$C000     メイン側からのデータを $C000からストアする
53E8 7F   D3FE          SXFER1 CLR  $D3FE      フラグレジスタクリア
53EB B6   D40A                LDA   $D40A      Ready !
53EE B6   D3FE          SXFER2 LDA  $D3FE      コマンド?
53F1 27   FB    53EE          BEQ   SXFER2     No
53F3 B7   D40A                STA   $D40A      Busy !
53F6 4A                       DECA             転送完了?
53F7 27   07    5400          BEQ   SXFER3     Yes
53F9 B6   D3FF                LDA   $D3FF      メイン側からのデータGET
53FC A7   80                  STA   ,X+        サブ側にストア
53FE 20   E8    53E8          BRA   SXFER1
5400 7E   C000          SXFER3 JMP  $C000      $C000へジャンプ
                        *
5403 10CE CFFF          SUBPRG LDS  #$CFFF     スタックポインタセット
5407 17   0097 54A1           LBSR  CLS        画面オールクリア
                        *  コマンドハンドラルーチン
540A 7F   D382          CMDH  CLR   $D382      コマンドレジスタクリア
540D B5   D40D                BITA  $D40D      INS LED ON
5410 B6   D40A                LDA   $D40A      Ready !
5413 B6   D382          CMDH1 LDA   $D382      コマンドあるか?
5416 27   FB    5413          BEQ   CMDH1      No
5418 B7   D40A                STA   $D40A      Busy !
541B B7   D40D                STA   $D40D      INS LED OFF
541E 80   E0                  SUBA  #$E0
5420 48                       ASLA
5421 30   8C 06               LEAX  <JMPTBL,PCR  ┐
5424 EC   86                  LDD   A,X          │
5426 AD   8B                  JSR   D,X          ├ 各コマンドをコール
5428 20   E0    540A          BRA   CMDH         ┘
                        *
542A      008F          JMPTBL FDB  CLS2-JMPTBL
542C      014B                FDB   GETKEY-JMPTBL
542E      0029                FDB   PRINT-JMPTBL
5430      00D0                FDB   EDSP-JMPTBL
5432      0028                FDB   RET1-JMPTBL
                        *  画面上のアドレス計算
5434 86   A0            CRTADR LDA  #160
5436 F6   D384                LDB   $D384      Get ´Y´ point
5439 3D                       MUL              D=B*160
543A 58   49                  ASLD             D=D*2
543C 58   49                  ASLD             D=D*2
543E 34   06                  PSHS  A,B
5440 F6   D383                LDB   $D383      Get ´X´ point
5443 1D                       SEX
5444 E3   E1                  ADDD  ,S++
5446 1F   01                  TFR   D,X
5448 39                       RTS
                        *  キャラクタパターンのアドレス計算
5449 C6   10            CHRADR LDB  #16
544B 3D                       MUL
544C 33   8D 0213             LEAU  CHRDAT,PCR
```

```
5450 33    CB               LEAU  D,U
5452 39            RET1     RTS
                   *
5453 108E  D387    PRINT    LDY   #$D387
5457 7D    D386    PRINT1   TST   $D386       表示終わり？
545A 27    F6   5452        BEQ   RET1        Yes
545C 7A    D386             DEC   $D386       文字数－1
545F 8D    D3   5434        BSR   CRTADR      表示位置計算
5461 7C    D383             INC   $D383       ┐
5464 7C    D383             INC   $D383       ┘表示位置更新
5467 A6    A0               LDA   ,Y+         キャラクタコードGET
5469 8D    DE   5449        BSR   CHRADR      キャラクタパターンアドレス計算
546B 8D    02   546F        BSR   TFR         画面に表示
546D 20    E8   5457        BRA   PRINT1
                   *
546F 34    20      TFR      PSHS  Y
5471 108E  0008             LDY   #8
5475 B6    D385    TFR1     LDA   $D385       カラーコードGET
5478 34    02               PSHS  A
547A EC    C1               LDD   ,U++        キャラクタパターンGET
547C B5    D409             BITA  $D409       VRAM ACC. ON
547F 64    E4               LSR   ,S          青のビットON？
5481 24    02   5485        BCC   TFR2        No
5483 ED    84               STD   ,X          青のVRAMに転送
5485 64    E4      TFR2     LSR   ,S          赤のビットON？
5487 24    04   548D        BCC   TFR3        No
5489 ED    89 4000          STD   $4000,X     赤のVRAMに転送
548D 64    E0      TFR3     LSR   ,S+         緑のビットON？
548F 24    04   5495        BCC   TFR4        No
5491 ED    89 8000          STD   -$8000,X    緑のVRAMに転送
5495 B7    D409    TFR4     STA   $D409       VRAM ACC. OFF
5498 30    88 50            LEAX  80,X
549B 31    3F               LEAY  -1,Y
549D 26    D6   5475        BNE   TFR1
549F 35    A0               PULS  Y,PC
                   *    全画面の消去ルーチン
54A1 B7    D408    CLS      STA   $D408       CRT OFF
54A4 B5    D409             BITA  $D409       VRAM ACC. ON
54A7 4F                     CLRA
54A8 5F                     CLRB
54A9 1F    01               TFR   D,X         X=0
54AB ED    81      CLS1     STD   ,X++        ┐
54AD 8C    C000             CMPX  #$C000      │0000～$BFFFまでに00をストア
54B0 26    F9   54AB        BNE   CLS1        ┘
54B2 B7    D409             STA   $D409       VRAM ACC. OFF
54B5 B6    D408             LDA   $D408       CRT ON
54B8 39                     RTS
                   *    ゲーム面の消去ルーチン
54B9 8E    029A    CLS2     LDX   #640+26     画面右半分クリア用ポインタセット
54BC CE    0299             LDU   #640+25       左半分    〃
54BF 34    50      CLS21    PSHS  X,U         ポインタセーブ
54C1 C6    C0               LDB   #192        192ドット行クリア
54C3 B5    D409             BITA  $D409       VRAM ACC. ON
54C6 6F    84      CLS22    CLR   ,X          VRAM（青）クリア ┐
54C8 6F    89 4000          CLR   $4000,X      〃 （赤）クリア │右半分
54CC 6F    89 8000          CLR   -$8000,X     〃 （緑）クリア ┘
54D0 6F    C4               CLR   ,U          VRAM（青）クリア ┐
54D2 6F    C9 4000          CLR   $4000,U      〃 （赤）クリア │左半分
54D6 6F    C9 8000          CLR   -$8000,U     〃 （緑）クリア ┘
54DA 30    88 50            LEAX  80,X        ┐
54DD 33    C8 50            LEAU  80,U        ┘1ドット下の行へ
54E0 86    32               LDA   #50         ┐
54E2 4A            CLS23    DECA              │時間待ち
54E3 26    FD   54E2        BNE   CLS23       ┘
54E5 5A                     DECB              1列終了？
54E6 26    DE   54C6        BNE   CLS22       No
54E8 B7    D409             STA   $D409       VRAM ACC. OFF
54EB 35    50               PULS  X,U         ポインタ復帰
54ED 30    01               LEAX  1,X         ┐
54EF 33    5F               LEAU  -1,U        ┘次の列へ
54F1 8C    02B0             CMPX  #640+48     全列終了？
54F4 26    C9   54BF        BNE   CLS21       No
```

```
54F6 39                           RTS                      終わり
                           *  エイリアン表示ルーチン
54F7      D383             POINTR FDB    $D383             共有RAM上のエイリアンデータのGETポインタセット
54F9      00               VC     FCB    0                 何段目のエイリアン処理中か
                           *
54FA 10AE 8C F9            EDSP   LDY    POINTR,PCR        YレジスタにエイリアンデータのGETポインタセット
54FE E6   8C F8                   LDB    VC,PCR            Bレジスタに何段目処理かセット
5501 4F                           CLRA
5502 8D   19     551D      EDSP1  BSR    EDSP3             1段(11匹)表示
5504 4C                           INCA
5505 81   0B                      CMPA   #11
5507 26   F9     5502             BNE    EDSP1
5509 5C                           INCB                     次の段へ
550A E7   8C EC                   STB    VC,PCR
550D C1   05                      CMPB   #5                全段終了？
550F 26   07     5518             BNE    EDSP2             No
5511 6F   8C E5                   CLR    VC,PCR            1段目から
5514 108E D383                    LDY    #$D383            GETポインタリセット
5518 10AF 8C DB            EDSP2  STY    POINTR,PCR
551C 39                           RTS
551D 34   06               EDSP3  PSHS   A,B
551F 86   A0                      LDA    #160
5521 3D                           MUL
5522 58   49                      ASLD                     Aレジスタ＝X座標  0～11 から
5524 58   49                      ASLD                     Bレジスタ＝Y座標  0～5
5526 34   06                      PSHS   A,B               表示位置を計算し，
5528 E6   62                      LDB    2,S               そのアドレスをXレジスタにセット
552A 58                           ASLB
552B 58                           ASLB
552C 4F                           CLRA
552D E3   E1                      ADDD   ,S++
552F 8E   0A04                    LDX    #640*4+4
5532 30   8B                      LEAX   D,X
5534 A6   A0                      LDA    ,Y+               エイリアンデータGET
5536 C6   21                      LDB    #33               A＝エイリアンタイプから
5538 3D                           MUL                      エイリアンのグラフィックパターンのデータの
5539 33   8D 003F                 LEAU   EADR,PCR          アドレスをUレジスタにセット
553D 33   CB                      LEAU   D,U
553F A6   C0                      LDA    ,U+               色データGET
5541 C6   08                      LDB    #8
5543 8D   0A     554F      EDSP4  BSR    EDSP5
5545 8D   08     554F             BSR    EDSP5             エイリアン1匹表示
5547 30   88 4C                   LEAX   76,X
554A 5A                           DECB
554B 26   F6     5543             BNE    EDSP4
554D 35   86                      PULS   A,B,PC
554F 34   06               EDSP5  PSHS   A,B
5551 34   02                      PSHS   A                 色コードスタックにPUSH
5553 EC   C1                      LDD    ,U++              グラフィックデータGET
5555 B5   D409                    BITA   $D409             VRAM ACC. ON
5558 64   E4                      LSR    ,S                色コード(青)ON？
555A 24   02     555E             BCC    EDSP6             No
555C ED   84                      STD    ,X                VRAM(青)にライト
555E 64   E4               EDSP6  LSR    ,S                色コード(赤)ON？
5560 24   04     5566             BCC    EDSP7             No
5562 ED   89 4000                 STD    $4000,X           VRAM(赤)にライト
5566 64   E0               EDSP7  LSR    ,S+               色コード(緑)ON？
5568 24   04     556E             BCC    EDSP8             No
556A ED   89 8000                 STD    -$8000,X          VRAM(緑)にライト
556E 30   02               EDSP8  LEAX   2,X               ポインタインクリメント
5570 B7   D409                    STA    $D409             VRAM ACC. OFF
5573 35   86                      PULS   A,B,PC
                           *  キー入力要求ルーチン
5575 B6   D401             GETKEY LDA    $D401             キーデータ参照
5578 B7   D383                    STA    $D383             共有RAMにストア
557B 39                           RTS
```

はじめて握るはんだごて PART3
リレーとリアルタイムクロックを使ったオーディオタイマの製作 I

志 沢 雅 彦

第4号で製作したPIA基板を使って，FMにリレーとリアルタイムクロック(RTC)を接続して，オーディオタイマを作ってみましょう。今回はその1ハードウエア編，次回はソフトウエア編となります。リレーは欲ばって4つ付けて，4系統までのコントロールができるようにしました。また，リアルタイムクロックは，電池でバックアップを行い，FMの電源を切っても止まらないようにしました。リアルタイムクロックの回路とリレー制御の回路は，独立に使うことができるので，オーディオタイマとしてだけでなく，電源の自動ON/OFF一般に使うことができます。

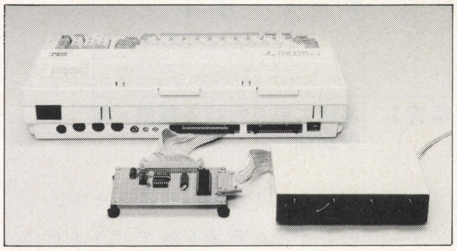


写真1 FM-7にPIAカードを介してRTCリレーユニットを接続

1. リアルタイムクロック(RTC)

リアルタイムクロック(RTC)とは，マイコンのための「時計」であって，プログラムで現在の時刻をCPUに読み込むことができます。基本的には，デジタル腕時計と同じ仕組みになっていて，出力先が液晶ディスプレイではなく，マイコンになったものです。もちろん水晶振動子を使ったものなので，正確さもデジタル時計と変わりません。

「なんだ，時計ならFMにはF-BASICでTIME$, DATE$関数があるじゃないか」と言う人がいるかもしれませんが，この関数は，FM本体内の水晶振動子を基準にしていて，CPUが割り込みを用いてサポートしているものなので，FMの電源を切ると時刻は失われ，電源を入れ直したときに，セットし直さなければなりません。また，パソコンに使われているCPUのクロック発生用水晶振動子は，一般にあまり精度が高くありませんし，「微調整」もできないので，長期にわたって動作させていると相当の狂いが生じます。

ところが，今回使用したリコーのRP5C01というICは，液晶時計などに使われているLSIと同じく低消費電力であるCMOS型なので，電池で動作させることが可能です。したがって，FMの電源をOFFにしても，RTCには電池から電源を供給するようにしておけば，FMの電源を入れるたびに時刻を設定する必要がなくなります。さらに，使っている水晶振動子も液晶時計に使っているものと同じで，微調整ができるように回路を組んだので，調整すれば高精度の時計にすることができます。

正確さの面だけではなく，機能の面でも，うるう年も判別するカレンダ機能やアラーム機能など，F-BASICによる時刻とは比べものになりません。

ほかにも，電池によってバックアップされるRAMを内蔵しているので，プログラムでこのRAMにデータを書き込んでおけば，FMの電源を切ってもその内容は変わらないので，いろいろと応用できます。

2. リレー

今回は，リレーをコンピュータの出力装置として使っています。リレーとは，コイル(電磁石)に電流を流すことにより接点を接触させて電流のON/OFFを行うものです(図3参照)。

今回のように，AC電源をON/OFFする場合のように制御する側と制御される側を電気的に絶縁する必要があるときは，リレーのような機械的な装置を用いたり，フォ

図1 本機のブロック図

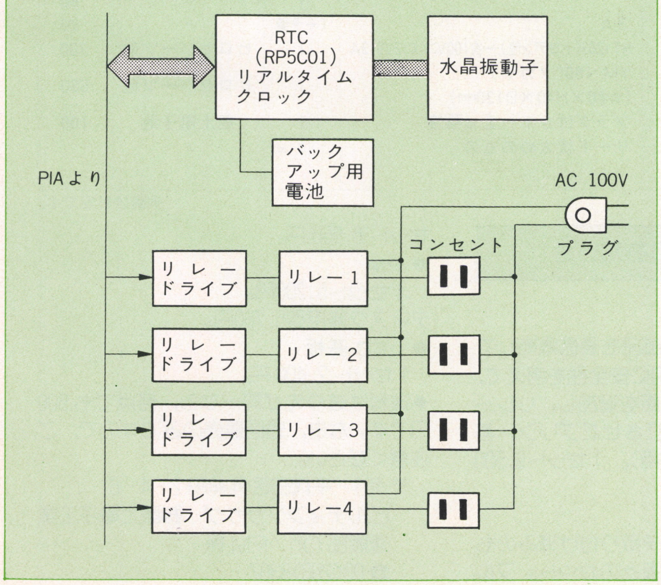
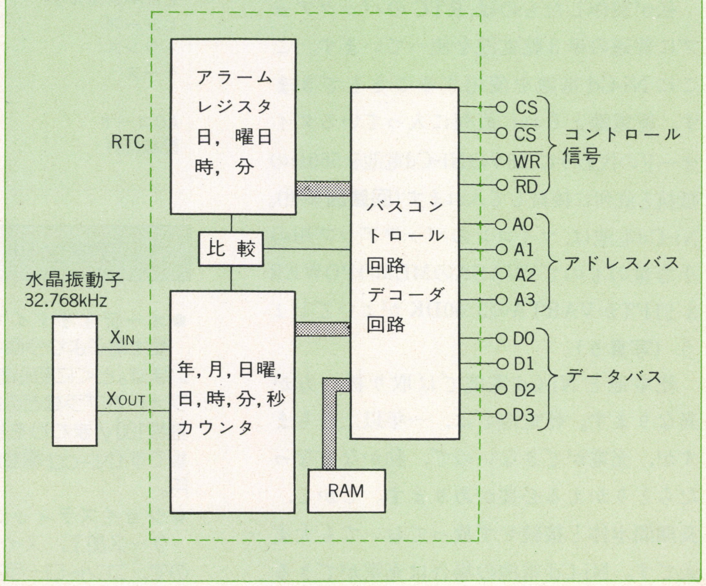


図2 RTCの構成



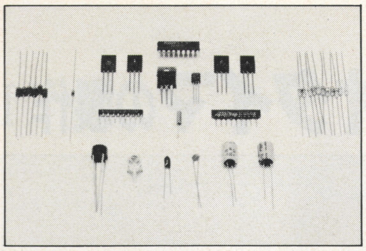
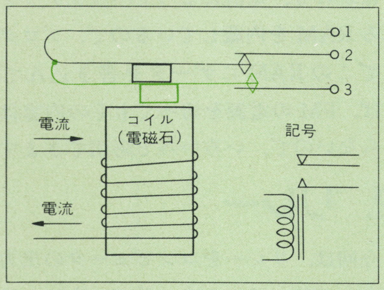


写真2　基板に取り付ける電気・電子部品

トカプラのように光を使ったりします。

リレーのコイルに電流を流すドライバには，トランジスタを使っています。これは，このリレーのコイルにかなり大きな電流(120mAぐらい)が流れるので，TTLでは荷が重すぎるためです。

図3　リレーの動作(模式図)



3. 部品集め

部品リストを**表1**に示します。「部品について」の項をよく読んでから集めてください。

今回は，抵抗，トランジスタなど細かい部品が多いので「個数」に注意してください。

私が製作したものは，RTCのバックアップに普通の単3乾電池を使っています。ここにNi-Cd電池を使用することもできます。乾電池とそれに直列に入っているダイオードの代わりに3.6VのNi-Cd電池を560Ωの抵抗と直列に接続して入れます(**回路図**参照)。Ni-Cd電池は，マイコンのバックアップ用の小容量のものでYUASAのMEMOPOWER 3-51FTかVARTAの3/60DKがよいでしょう(**写真5**)。

乾電池とNi-Cd電池では取り扱い方が異なります。乾電池では，一年以上もちますが，充電ができないので，動かなくなったらとりかえる必要があります。しかし，長期間本体と接続せず放っておいても大丈夫です。Ni-Cd電池の場合は充電ができる

表1　部品リスト

品名	型番(メーカー名)	個数	販売店	単価
LSI	RP5C01(リコー)	1	亜土電子他	950
ICソケット	18P(バーンディ他)	1	〃	60
トランジスタ	2SD794A(NEC)	4	〃	60
	2SA1015	1	〃	20
ICレギュレータ	7812	1	〃	160
ダイオード	1N4002	6	〃	15
シリコンブリッジダイオード	W02M200V1.5A	1	〃	70
ツェナーダイオード	RD3.9BE400mW(NEC)	1	〃	40
LED	ブラケット付き発光ダイオード	4	〃	80
リレー	NB-1 AE-8211(松下) AC250V 5A 家電用 12V用リレー	4	ヒロセパーツ他	340
水晶振動子	32.768kHz	1	亜土電子他	400
集合抵抗	100kΩ×8	1	〃	50
〃	4.7kΩ×8	1	〃	50
抵抗	3.3kΩ1/8W	4	〃	10
〃	820Ω1/8W	4	〃	10
〃	100kΩ1/8W	1	〃	10
〃	1kΩ1/8W	1	〃	10
〃	4.7kΩ1/8W	1	〃	10
コンデンサ	タンタル4.7μF16V	1	〃	50
	セラミック27pF	1	〃	10
	ケミコン100μF25V	1	〃	30
トリマコンデンサ	30pF	1	〃	40
電池ホルダ	単3×2用	1	〃	50
乾電池	単3	2	〃	
電池コード		1	〃	20
ヘッダ	26pフラットケーブル用(山一，ヒロセ他)	1	〃	370
基板	AD0725G(亜土電子)	1	〃	700
スペーサ	1cm長ネジ付	4	〃	30
コンセント	AC-1F (EDK)	4	〃	80
ヒューズホルダ		1	〃	80
ヒューズ	3A	1+予備	〃	20
トランス	HT1203(トヨデン他)一次100V二次12V0.3A	1	ヒロセパーツ他	700
ケース	YM-180(タカチ)(W80×H40×D130mm)	1	ヒロセパーツ他	680
ACコード	プラグ付き1～2m程度	1	亜土電子他	100
配線材料	リード線太めのもの	1m	〃	

単価は目安です。

部品セットのお知らせ

● **オーディオタイマ**
亜土電子より今回の部品を通信販売しています。とくに初心者用に安全性を考えて，フォトカプラなどの回路を追加し，リレー基板100Vまわりを独立させたプリント基板です(Ni-cd電池使用)。1セット9,500円。

● **ジョイスティック**
ケース加工，スイッチ取り付け済みのもので，コントローラ基板とのセット。フルセット6,650円。

● **PIA**
1セット5,200円。
ディスク兼用型5,700円。

● **LED基板**
1セット2,400円。

●送料は送り先が同一なら，何点でも500円です。なお，通信販売のみですので，ご注意ください。

〒101　千代田区外神田2-4-6
ビルディングササゲ　㈱亜土電子工業
通販部 Oh!FM係
☎03(253)8307

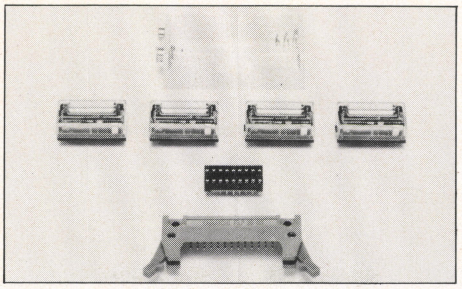


写真3　基板に取り付けるソケット類とリレー

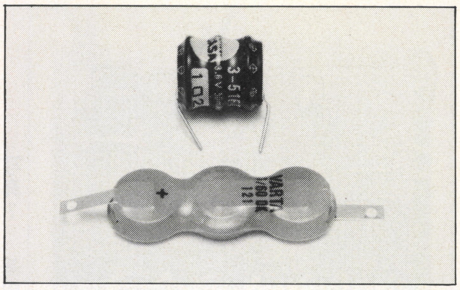


写真5　Ni-Cd電池

ので，交換の必要はないのですが，自己放電が大きいので，長期間放置すると，たとえはじめに満充電の状態でも1か月ぐらいでなくなってしまいます。したがって，2週間以上放置されるようなことがないようにしなければいけません。

4. 製　作

図4のフローチャートにしたがって，進めてください。

(1)　基板の製作

①ヘッダ，ICソケット，リレーの取り付け

ヘッダ，ICソケット，リレーを基板に差し込んではんだづけをします。

②トリマコンデンサの取り付け

トリマコンデンサは，基板の裏（銅はくの付いている側）にはんだづけします。

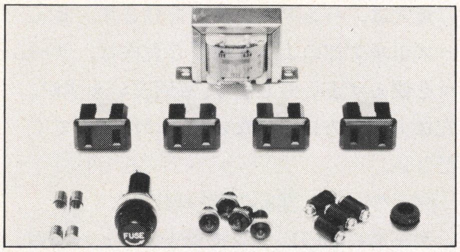


写真4　ケースに取り付ける部品

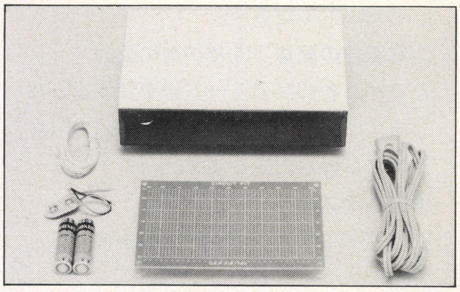


写真6　その他の部品

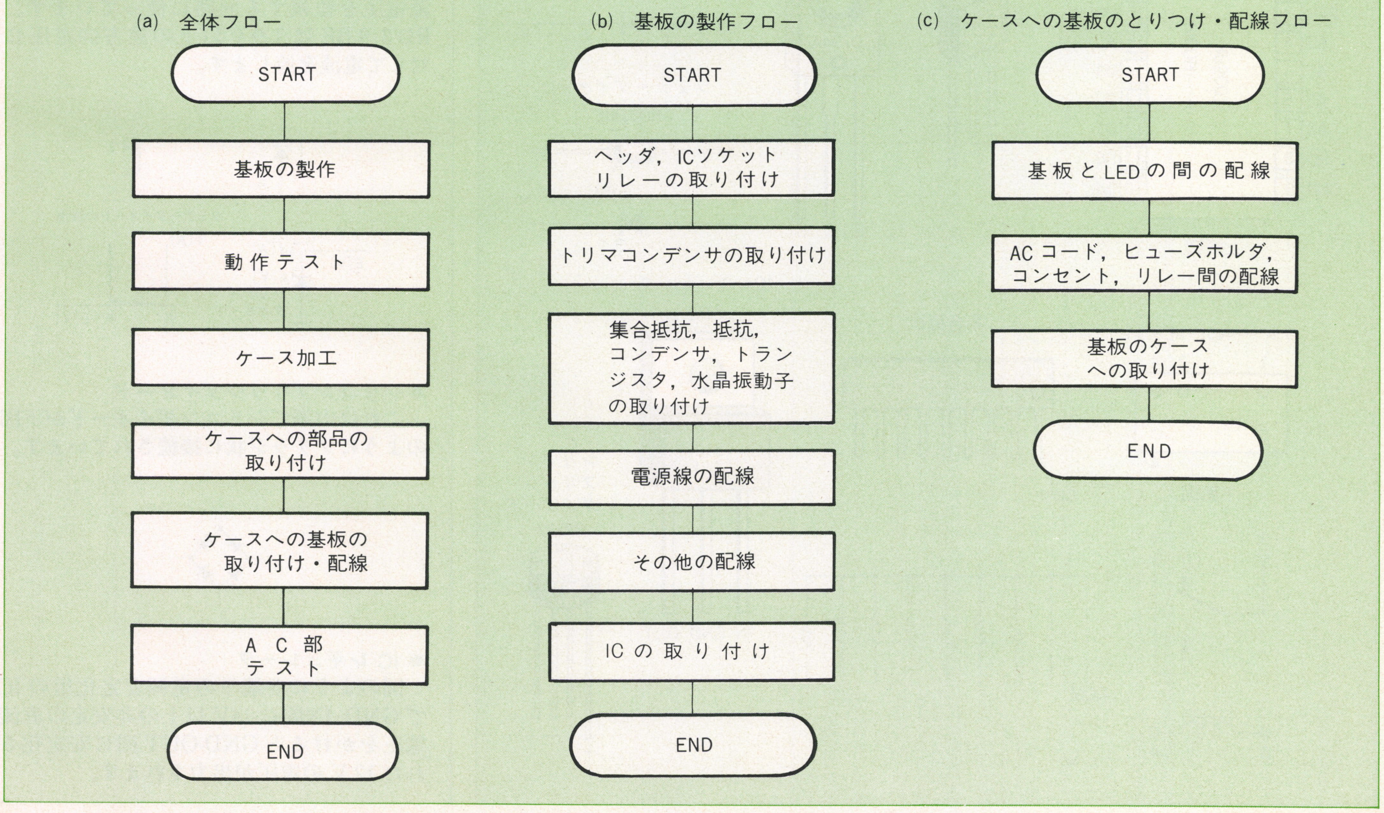
トリマコンデンサは，図6のように，まず3つの足を外側に曲げておいて，銅はくの上に端子がくるように置いてから，端子の上にかぶせるようにしてはんだを流します。

③集合抵抗，抵抗，コンデンサ，トランジスタ，水晶振動子の取り付け

集合抵抗は，白い印が付いているのがコモン端子です。タンタルコンデンサ，トランジスタは極性に充分注意して取り付けてください。リード線は折り曲げて有効に使えば，テフロン線による配線を減らすことができます（写真7参照）。

水晶振動子は，リード線が細いので，強い力を与えないようにしてください。また熱にも弱いので，リード線は長めに残

図4　製作フローチャート



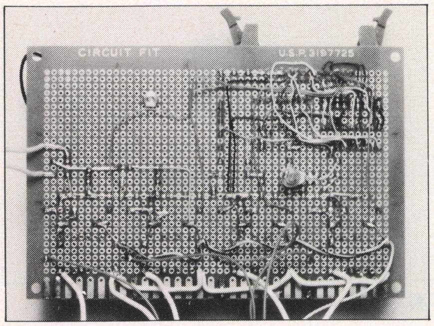


写真7 基板裏

しておいてください。

④電源線の配線　⑤その他の配線

なるべく各部品のリード線を使うようにしますが，それで配線できないところは，テフロン線で配線します。

⑥動作テスト

後に述べるテストプログラムを使って，動作テストを行ってください。

(2) ケース加工

ケースは，**図9**を参考に自分でレイアウトを考えて作ってください。その際，部品同士が接触しないように気を付けてください。

角穴は，ハンドニブラであけます。ヒューズホルダ部のような大きな丸穴は，リーマを使えばよいでしょう。毎回言いますが，穴は大きくあけすぎないように注意してください。

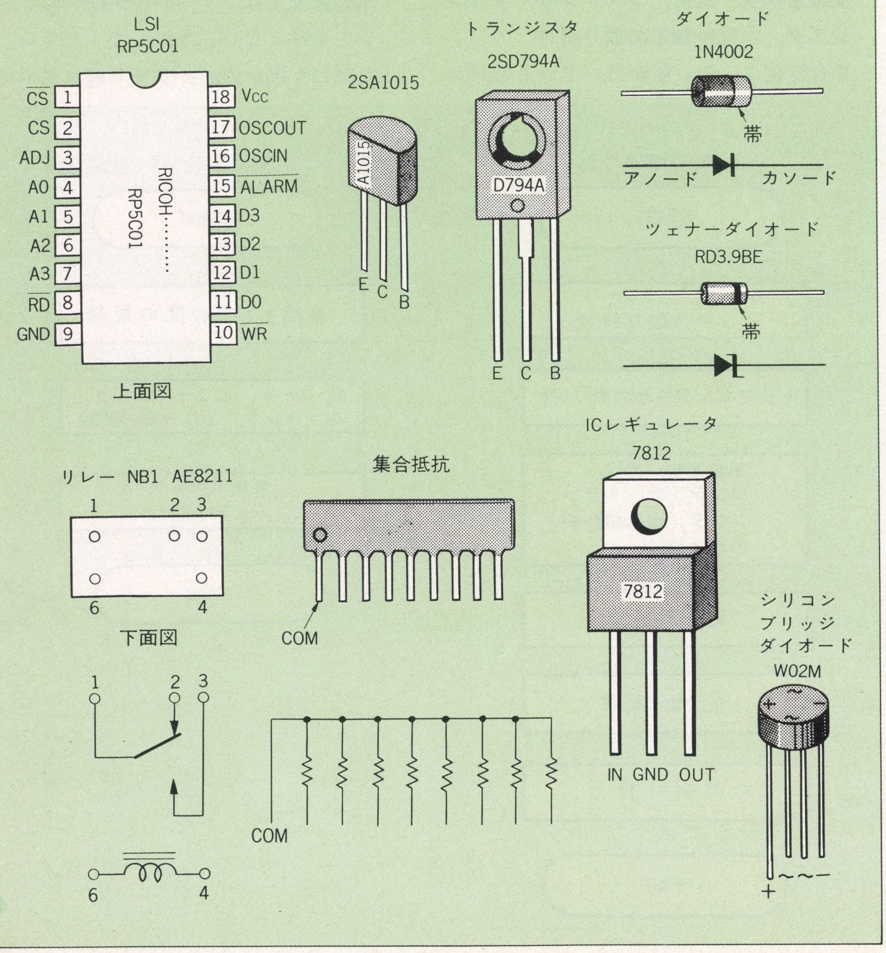
(3) ケースへ部品の取り付け

ケースにLED，コンセント，ヒューズホルダ，ゴムブッシュを取り付けます。

(4) 基板をケースに取り付ける

LEDと基板（リレー）との間は細い線でも構いませんが，ACコンセントと基板(リレー）との間の配線は太めのものを使ってください。

図5 ピン接続図



部品について

●リアルタイムクロック（RTC）

RTCはリコーのRP5C01です。このICはCMOS型なので，静電マットかアルミホイルに包んで保存しておいてください。

●トランジスタ

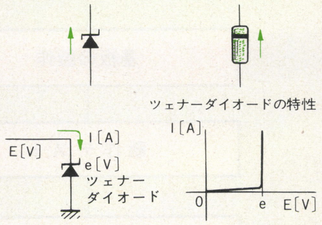
トランジスタ（2SD794A）はプラスチックパッケージに入った中出力のもので，リレーの制御に使います。2SA1015は小信号用のものです。

●ダイオード

シリコンダイオードで，IN4002という整流用のもので100V1Aのものを使いました。

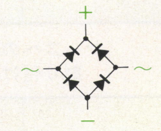
●ツェナーダイオード

ツェナーダイオードは定電圧ダイオードとも呼ばれ，一定電圧以上の逆方向電圧をかけると電流が流れるもので，一般に電源の安定化回路に使われますが，今回の製作では，FMの電源ON，OFF時の過渡状態において，RTCが誤動作しないように，電源電圧を監視する回路に使っています。RD3.9BEは3.9V以上の逆方向電圧に対して電流を流します。



●シリコンブリッジダイオード

これは内部でシリコンダイオードが下図のようにブリッジ状に接続されています。



● ICレギュレータ

図のように3端子の電源安定化用のICでGND-IN間に14V以上の非安定の直流電圧をかけるとGND-OUT間に安定化された12Vの電圧が出力されます。

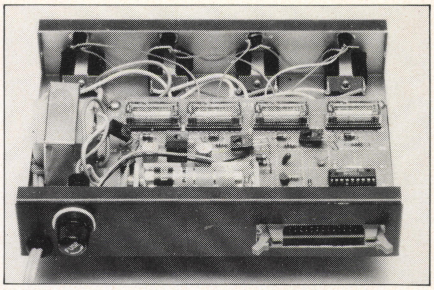
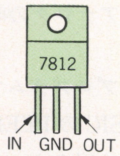


写真8　ふたをあけたところ



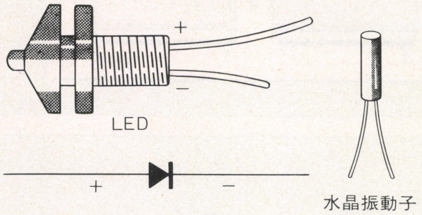
●トランス

リレーへの電源供給用に12Vが必要なのでAC100Vからトランス12Vに落としています。このトランスは，一次側100V，二次側12V 0.3Aを使いましたが，これはケースの大きさが小さくて，これより容量の大きいものだと入らなかったために使ったので，少々容量が不足するようですので二次側が12V1Aぐらいのものを使うほうがよいでしょう。

● LED

LEDはケースに取り付けるので，裸のものではなくプラスチックのブラケットに入ったものにしました。ブラケットに入ったLEDは各種ありますから，自分の好みに合ったものにしてください。ただし，私が製作した回路では，赤のLEDを使ったので，それに合わせてR6～R9を820Ωとしてありますが，緑や黄色のものにした場合，多少暗いかもしれません。そのときはR6～R9を600Ωぐらいにするとよいでしょう。

また，LEDによって極性の表し方が異なります。私が使用したものは，ケースに+，−の表示がありましたが，一般に，線の太いほうあるいは長いほうが+です。

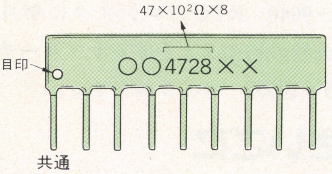


●水晶振動子

これは時計用によく使われている32.768 KHzのものを使います。小さな円筒型のアルミ缶に入っています。

●集合抵抗

細長い板からリード線が9本出ているもので，8本の抵抗が入っています。各抵抗の片側はすべて内部でつながっていて，1本にまとまって出ているものです。100 kΩ，4.7 kΩを使います。

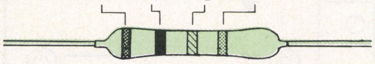


●抵抗

1/8Wのごく普通のカーボン抵抗です。

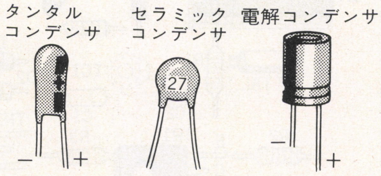
抵抗のカラーコード

				±5 ±10 ±20%
3.3KΩ	橙 3	橙 3	赤 2	金or銀or無印
820Ω	灰 8	赤 2	茶 1	〃
100KΩ	茶 1	黒 0	黄 4	〃
1KΩ	茶 1	黒 0	赤 2	〃
4.7KΩ	黄 4	紫 7	赤 2	〃



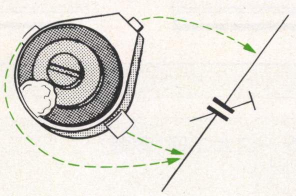
●コンデンサ

タンタルコンデンサとセラミックコンデンサは，どちらも第5号で使ったものと同じ形状です。



●トリマコンデンサ

これは，容量の調節ができるコンデンサで，水晶の発振周波数の微調整に使います。

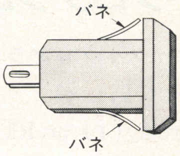


●電池ホルダ，電池コード

プラスチックでできた，よくトランジスタラジオに使われているものです。ホルダから直接リード線が出ているものでもかまいません。

●コンセント

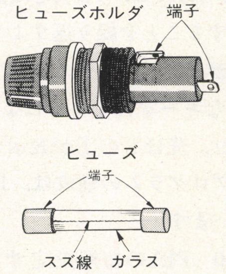
コンセントは，ケースにネジ止めなどで簡単に取り付けられるものにしてください。



私が使用したものは，長方形の穴に押し込むだけで，プラスチックのバネで止まるようになっていました。

●ヒューズホルダ，ヒューズ

ヒューズホルダは円筒型のもので，中に入れるヒューズは円筒ガラスに入ったものです。ヒューズには，長いものと短いものがありますが，どちらでも構いません。私は長いものを使いました。ただし，もちろん，ヒューズホルダとヒューズは同じ規格のものでなければいけません。



●ケース

ケースは，アルミ製のW 180×H 40×D 130 mmの大きさのものを使いました。正面と背面が鉄板でできたものもありますが，これは穴あけが大変です。アルミ製のものにしたほうがよいでしょう。大きさは，これより少々大きめのほうがよいと思います。私が製作したものは，かなりつめこみになってしまいました。

●配線材料

AC100V側の配線には大電流(3A)が流れる可能性がありますから，テフロンワイヤではなく，太めのビニール線(20芯ぐらい)を使います。

●基板

リレーの接点にAC100Vがかかりますので，絶縁度の高いガラスエポキシ製を使ってください。

●リレー

AC250V 5Aの家電用のものを用いました。直流の12Vで動作するものです。オーディオタイマに用いるなら充分でしょう。

●その他

その他の部品については，4号・5号を参照してください。

5.テストプログラム

リスト1にテストプログラムを示します。RUNさせると，

relay(1) or RTC(2)……①

と聞いてきますから，1または2を入力してリターンを押してください。

1にするとリレーのテストを行います。LEDが左から右へ，順に流れるようにONになります。そのとき同時に，リレーのカチッという音がするはずです。

2にするとRTCのテストを行います。84年1月1日日曜日0時0分0秒を初期設定したのち，RTCから日付と時間を読み出し，表示することを繰り返すようになっています。秒がインクリメントされていくはずです。適当なキーを押すことにより，テストが中断され，先ほどの①が表示されます。

テストプログラムの構成は，以下のようになっています。

1027～1190　PIAの初期設定サブルーチン
1200～1310　リレーのテストサブルーチン
4000～4190　RTCのレジスタへ4ビットのデータを書き込むサブルーチン
5000～5120　RTCのレジスタから4ビットのデータを読み込むサブルーチン
6000～6035　RTCのモードの初期設定サブルーチン
6037～7120　RTCのレジスタ(タイマレジスタのみ)を表示するサブルーチン
8000～8090　RTCのテストを行うサブルーチン
9000～9040　RTCのレジスタに年月日時分秒の初期設定をするサブルーチン
10000～10050　メインルーチン

6.さいごに

次回は，今回製作したRTC＋リレーの装置を，オーディオタイマとして使うためのプログラムを発表します。日付から分までを指定できるもので，データは20個まで定義できるようになっています。また，画面にはデジタルとアナログの時計が表示されます。

今回のテストプログラムでは，RTCへのアクセスはBASICのPEEK，POKEを使っていますが，次回のオーディオタイマでは機械語を使います。

参考文献

トランジスタ技術　1983.1　P.268—「リアルタイム・クロック」

図7　基板上面部品配置

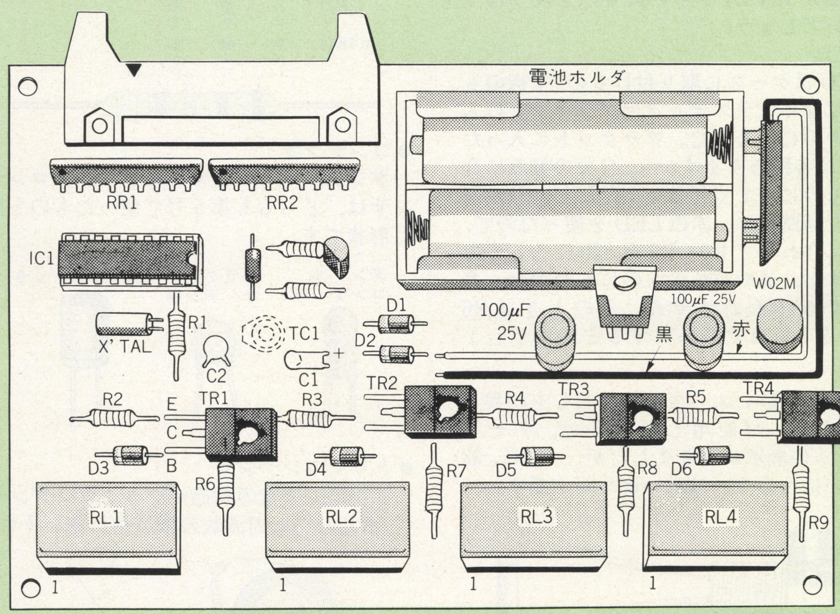


図6

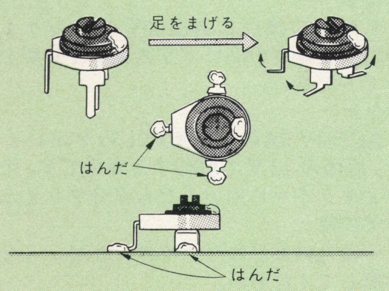
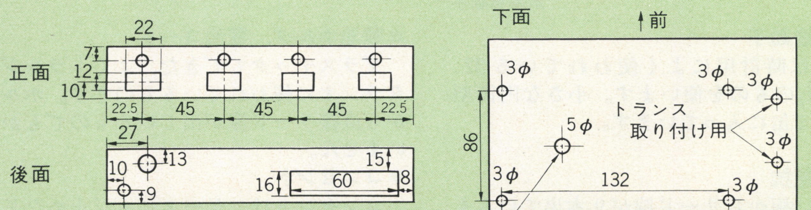
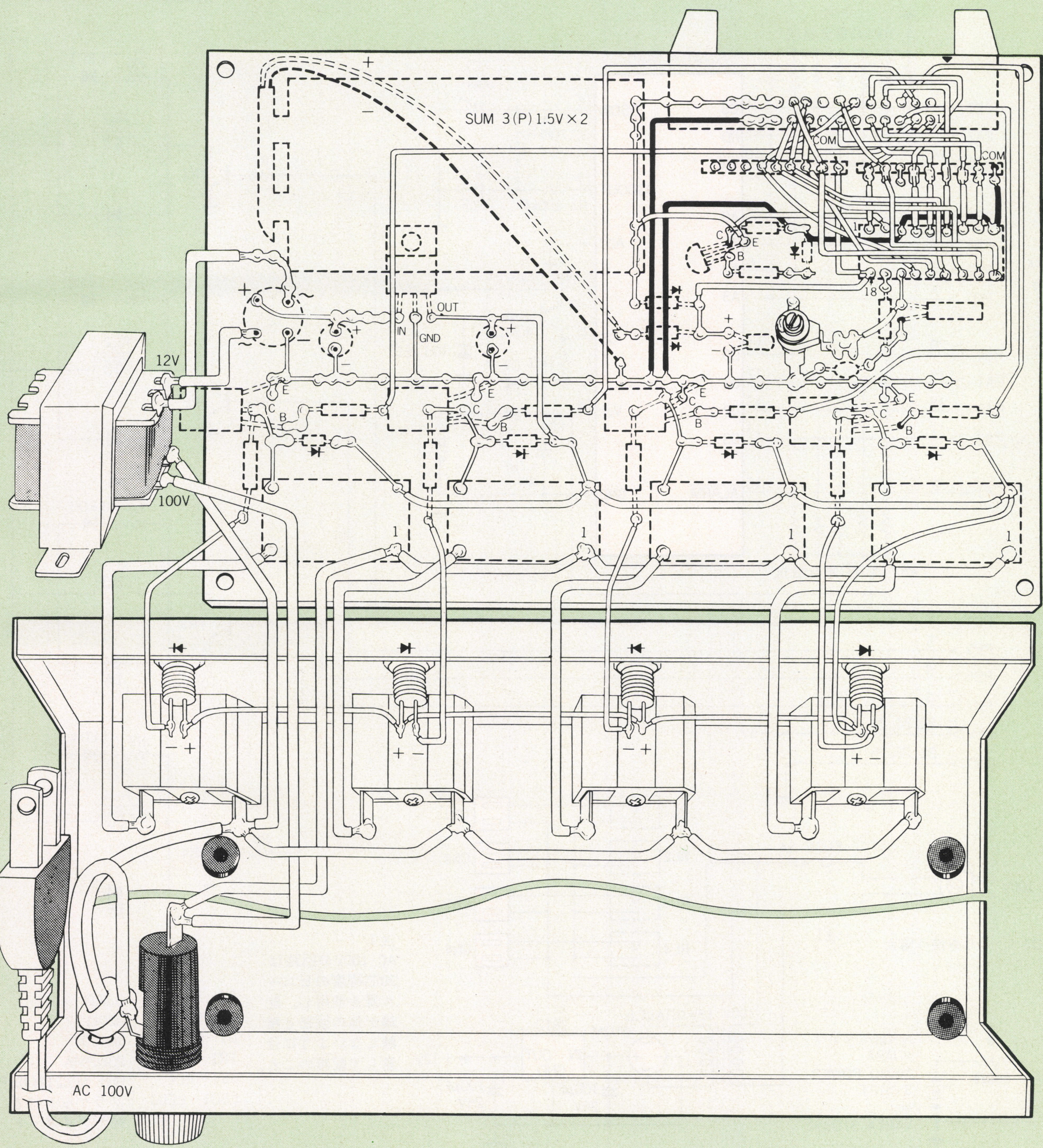


図8　ケース加工図(参考)

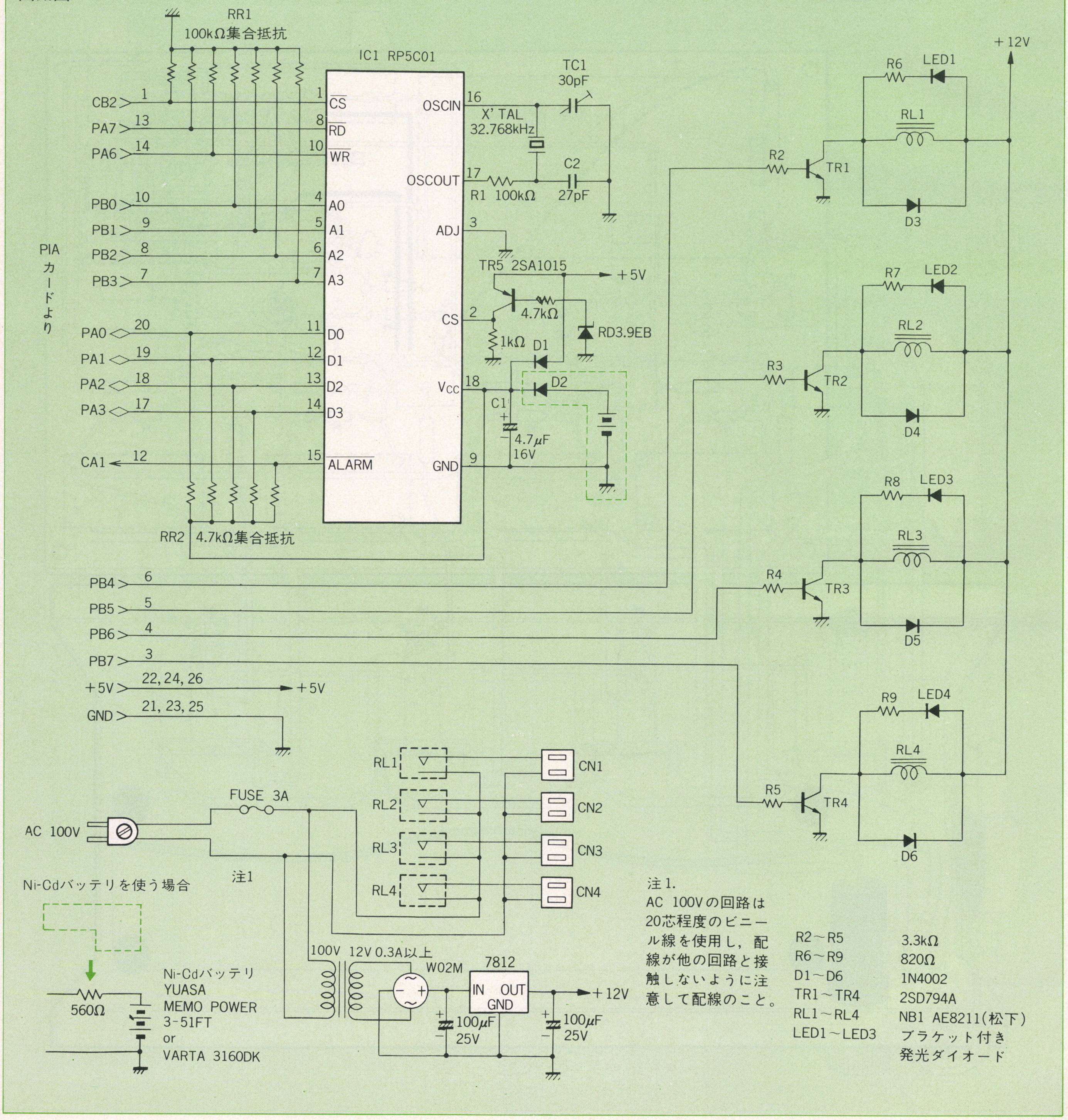


※　使用する部品が異なれば当然寸法も変わるので注意してください。

図9　ケースとの実体配線図



回路図



注1.
AC 100Vの回路は20芯程度のビニール線を使用し，配線が他の回路と接触しないように注意して配線のこと。

R2～R5	3.3kΩ
R6～R9	820Ω
D1～D6	1N4002
TR1～TR4	2SD794A
RL1～RL4	NB1 AE8211(松下)
LED1～LED3	ブラケット付き発光ダイオード

リスト1 テストプログラム

©1983 M.SHIZAWA

```
1000 ' Oh!-FM 1984 APRIL relays and RTC test program
1010 '   v.1.0 1983/11/25   by M.Shizawa  All right reserved
1020 '
1021 GOTO 10000
1022 '
1027 '******* pia setup ********
1030 ' PIA resister definition
1040 PIA = &HFDFC
1050 DDRA = PIA + 0: PIRA = DDRA: CRA = PIA + 1
1060 DDRB = PIA + 2: PIRB = DDRB: CRB = PIA + 3
1070 '
1080 ' initialize PIA
1090 POKE CRA, &H04: POKE PIRA, &HFF 'avoid trangent state (control line high)
1100 'PIRA mode         PIRA=1111 1111
1110 POKE CRA, &H00: POKE DDRA, &HC0: POKE CRA, &H04
1120 'DDRA mode         DDRA=1100 0000   CRA=xx000100 (x=0)
1130 '
1140 POKE CRB, &H04: POKE PIRB, &H00 'avoid trangent state (relay off)
1150 'PIRB mode         PIRB=0000 0000
1160 POKE CRB, &H00: POKE DDRB, &HFF: POKE CRB, &H34
1170 'DDRB mode         DDRB=1111 1111   CRB=xx1101xx (x=0)
1180 RETURN
1190 '
1200 '****** relay test ******
1210 WHILE INKEY$=""
1220   FOR RELAYNO=0 TO 3
1230     POKE PIRB, 2^(RELAYNO+4)
1240     GOSUB 1300
1250   NEXT RELAYNO
1260 WEND
1265 RETURN
1280 '
1290 '****** wait ******
1300 FOR I=0 TO 100: NEXT I
1310 RETURN
4000 '
4010 '*******  write to RTC register *********
4020 POKE CRB, PEEK(CRB) AND &HF7 'assert *cs
4030 POKE PIRB, (PEEK(PIRB) AND &HF0) OR A% 'write address
4040 POKE CRA, PEEK(CRA) AND &HFB 'ddra
4050 POKE DDRA, PEEK(DDRA) OR &H0F 'pa0-3 out
4060 POKE CRA, PEEK(CRA) OR &H04 'pira
4070 POKE PIRA, (PEEK(PIRA) AND &HE0) OR D%  'assrt *wr
4080 POKE PIRA, PEEK(PIRA) OR &H40 'negate *wr
4140 POKE CRA, PEEK(CRA) AND &HFB 'ddra
4150 POKE DDRA, PEEK(DDRA) AND &HF0 'pa0-3 in
4160 POKE CRA, PEEK(CRA) OR &H04 'pira
4170 POKE CRB, PEEK(CRB) OR &H08 'negate *cs
4180 RETURN
4190 '
5000 '****** read from RTC register ********
5020 POKE CRB, PEEK(CRB) AND &HF7 'assert *cs
5030 POKE PIRB, (PEEK(PIRB) AND &HF0) OR A% 'write address
5040 POKE CRA, PEEK(CRA) AND &HFB 'ddra
5050 POKE DDRA, PEEK(DDRA) AND &HF0 'pa0-3 in
5060 POKE CRA, PEEK(CRA) OR &H04 'pira
5070 POKE PIRA, PEEK(PIRA) AND &H7F 'assert RD
```

```
5080 D%=PEEK(PIRA) AND &H0F ´read data
5090 POKE PIRA, PEEK(PIRA) OR &H80 ´negate RD
5100 POKE CRB, PEEK(CRB) OR &H08 ´negate *cs
5110 RETURN
5120 ´
6000 ´****** RTC initialize ********
6010 A%=&HD: D%=&H8: GOSUB 4000 ´mode reg(1000,timer on,alarm off,mode 00)
6020 A%=&HE: D%=&H0: GOSUB 4000 ´test reg(0000,not test mode)
6030 A%=&HF: D%=&HF: GOSUB 4000 ´reset reg(1111,1hz off,16hz off,time,alm res)
6035 RETURN
6036 ´
6037 ´******* print registers ********
6040 A%=&HD: D%=&H0: GOSUB 4000 ´timer off
6050 FOR ADDR%=0 TO &HC ´read all registers
6060   A%=ADDR%: GOSUB 5000: DD%(ADDR%)=D%
6070 NEXT ADDR%
6075 A%=&HD: D%=&H8: GOSUB 4000 ´timer on
6076 J%=&HB:GOSUB 7000:PRINT "年 ";
6080 J%=&H9:GOSUB 7000:PRINT "月 ";
6090 J%=&H7:GOSUB 7000:PRINT "日 ";
6095 GOSUB7100 ´print date
6100 J%=&H4:GOSUB 7000:PRINT "時 ";
6110 J%=&H2:GOSUB 7000:PRINT "分 ";
6120 J%=&H0:GOSUB 7000:PRINT "秒 ";
6130 PRINT
6140 RETURN
6150 ´
7000 ´******* print 2 count *********
7010 PRINT CHR$(DD%(J%+1)+&H30);CHR$(DD%(J%)+&H30);
7020 RETURN
7030 ´
7100 ´******* print date **********
7110 PRINT MID$("sunmontuewedthufrisat",1+3*DD%(&H6),3);" ";
7115 RETURN
7120 ´
8000 ´******* test RTC ********
8010 GOSUB 6000 ´initialize RTC
8015 GOSUB 9000 ´set registers
8020 WHILE INKEY$=""
8030   GOSUB 6037 ´print registers
8040 WEND
8050 RETURN
8090 ´
9000 ´****** initialize date,time ********
9001 RESTORE 9040
9002 FOR A%=0 TO &HC
9010   READ D%: GOSUB 4000
9020 NEXT A%
9030 RETURN
9040 DATA 0,0,0,0,0,0,0,1,0,1,0,4,8
10000 ´
10010 ´****** main routine ******
10015 DIM DD%(&HC)
10020 GOSUB 1030
10025 WHILE 1
10030   INPUT "relay(1) or RTC(2)",A
10035 PRINT "hit any key to stop."
10040   ON A GOSUB 1200,8000
10050 WEND
```

本プログラムのBASICチェックサム(p. 41参照)

LINE	+0	+1	+2	+3	+4	+5	+6	+7	SUM
1000	00	00	00	2A	00	00	00	98	:C2
1050	CE	D6	00	00	BE	00	CD	00	:2F
1130	00	94	00	EC	00	E0	00	00	:60
1210	5A	44	E2	44	59	2E	E0	00	:2B
1290	00	93	E0	00	00	3D	3A	46	:30
4050	8C	F0	37	9C	46	BE	F0	F6	:39
4180	E0	00	00	3D	3A	46	BE	F0	:4B
5070	E7	FF	A0	F6	E0	00	00	5A	:B6
6013	4F	59	52	69	E0	00	00	51	:94
6050	E8	80	7F	59	F1	E9	E8	48	:4A
	B2	09	6A	EB	48	38	7D	B7	/C4

LINE	+0	+1	+2	+3	+4	+5	+6	+7	SUM
6100	E6	E5	E4	8D	E0	00	00	D5	:F1
7020	E0	00	00	60	E0	00	00	46	:66
8015	18	5A	50	2E	E0	00	00	F1	:C1
9002	0E	03	A5	E0	C0	00	00	89	:DF
10020	44	AA	29	BA	15	2E	00	00	:14
	30	EC	02	B5	75	2E	00	95	/0B

for FM-7,8,11

使い勝手のよいマシン語モニタ

高橋　良之

はじめに

日頃FMシリーズを使ってマシン語プログラムを作成する場合，そのデバッグ（いわゆる虫取り）に困っていませんか。F-BASIC付属のモニタには必要最少限の機能しかなく，デバッグに骨が折れるうえ，お世辞にも使いやすいとはいえません。

そこで今回，F-BASICモニタの拡張版ともいうべきものを製作しましたのでご紹介いたします。これは，富士通から販売されている「デバッガ」の機能をすべて包含し，さらに初心者にもはるかに使いやすいよう，拡張されています。ただし，その代償としてメモリサイズが大きくなりましたが，勘弁してください。

BASICの勉強中でマシン語はまだという方が，単に雑誌のマシン語のダンプリストを打ち込むツールとして利用しても，絶大な力を発揮するはずですのでご利用ください。

特　徴

各機能ごとの詳細は後述しますが，主な特徴としては次の点が挙げられます。

①スクリーンエディタ風にメモリへの書き込み，エディットができる。16進形式，ASCII形式いずれでもエディットでき，1バイト書き込むごとに縦横のチェックサムがリアルタイムに更新表示されるので，マシン語ダンプリストの打ち込み，確認，修正が非常に楽になること請け合いです。もちろん，16進はテンキーから打ち込めますし，オプションにより上下にスクロールさせることも可能です。

②そのレベルごとに必要とされるコマンドが，必要に応じてスクリーンの最下段にグリーンで表示されるなど，操作性と使いやすさを考慮した，ユーザフレンドリな設計となっています。ともかく一度，走らせてみてください。

③6809マシン語では当然ですが，完全なポジションインディペンデントコードで構成されているため，フリーエリアなら，どのメモリ位置からでもロードが可能です。デバッグ対象マシン語プログラム，打ち込みダンプリストアドレスなどを考慮して，実際のロードアドレスを定めてください。

④また，F-BASICのインタプリタ内ルーチンは一切使用していませんので，将来BASICが変更されても，BIOSとサブシステムのインタフェース仕様が変わらないかぎり，そのまま動くはずです。そのため汎用性のある下位レベルの基本ルーチンを自作せざるをえないハメになり，その部分だけで約2Kバイト弱も費やしてしまいましたが……。せっかくですから次号以降で，その部分のエッセンスのみをソースリスト付きで公開する予定です。

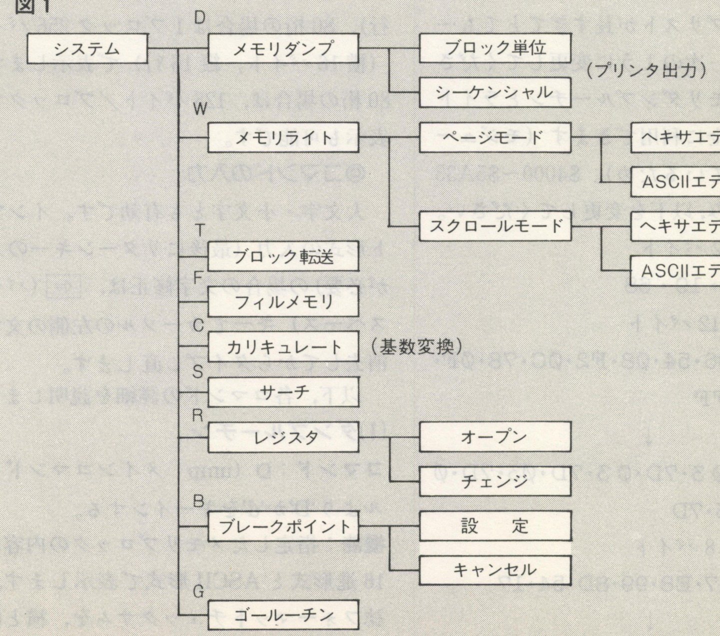
⑤前述の③，④から，賢明な読者ならもうお気付きと思いますが，FM-7の場合，当システムを裏RAMにおいて裏RAM上で実行できますので，32Kバイト以上のフリーエリアをフルに活用することができます。

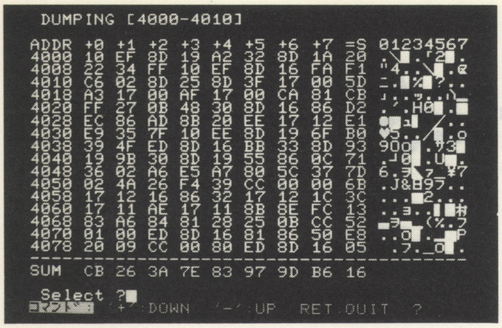
（当システムの構成については，図1を参照してください。）

ダンプリストの打ち込み方

まず，念のため，CLEAR,&H3FFFを実行します。次にVol.3（あるいはFILES Oh!FM）のチェックサムプログラムで打ち込むか，MONコマンドを実行し，F-BASICのモニタに入ります。MONコマンドの場合は，'*'のプロンプトが表示されま

図1





すので，M 4000を入力し，誤りのないように1バイトずつダンプリストどおり打ち込んでください。なにせプログラム自身，約10Kバイトもあるため打ち込みに苦労するでしょうが，あせらずじっくり打ち込んでください。

この次からは，このプログラムを使って打ち込めば，入力がはるかに楽になりますのでがんばってください。

なお，ダンプリストはFM-7，8用になっていますが，FM-11をお持ちの方は，指定部分の2バイトだけを置き換えてください。

打ち込みが終わっても，絶対すぐに走らせないでください。暴走したら，せっかくの努力が水の泡です。

まず，Vol.3のチェックサムプログラムを走らせてみて，誤りがないか入念にチェックしてください。OKならディスクかテープにすぐセーブしておいてください。

SAVEM"MONITOR", &H4ØØØ, &H68FF, &H4ØØØ⏎

また，ダンプリストが長すぎてとても……という方は，次のように変更してください。一応，メモリダンプルーチンとライトルーチンは完全に利用できます（モジュール構成をとっているため）。$4000～$5A33までを打ち込み，以下を変更してください。

・$40C5より2バイト
26・A4→19・68

・$56B4より12バイト
Ø5・ØF・Ø6・54・Ø8・F2・ØC・78・ØF・9F・1Ø・FF
↓
Ø3・7D・Ø3・7D・Ø3・7D・Ø3・7D・Ø3・7D・Ø3・7D

・$5A2Bより8バイト
8D・3F・17・E8・99・8D・54・17
↓
17・F8・EC・39・39・FF・ØØ・ØØ

$5A34以降は打ち込みません。

コマンドの説明

まず最初に，各コマンド共通の機能について説明します。

◎CTRL-Cキーの機能

キー入力待ちの状態およびダンプエディット（例外：ASCIIモードエディット中では使用できない）中にキーインすることにより，無条件にメインコマンドレベルに戻ります（Abort）。途中である仕事を打ち切りたい場合などに使用してください。

◎ESCキーの機能

画面の表示を一時的に停止させます。ダンプを一時的に停止させたいときに使用すると便利です。

◎画面モード

40桁，80桁を自動的に判別します。つまり，ダンプエディット時に40桁の場合は1ブロック128バイト（横8バイト，縦16行），80桁の場合は1ブロック256バイト（横16バイト，縦16行）で表示します。80桁の場合は，128バイト／ブロックでの表示も可能です。

◎コマンドの入力

大文字・小文字とも有効です。インプット形式の入力（最後にリターンキーの入力が必要）の場合の文字修正は，⇦（バックスペース）キーでカーソルの左側の文字を消去してからタイプし直します。

以下，各コマンドの詳細を説明します。

(1)ダンプルーチン

コマンド：D（ump メインコマンドレベルより'D'か'd'をキーインする。

機能：指定したメモリブロックの内容を，16進形式とASCII形式で表示します。本誌フォーマットチェックサムを，横と縦の両方で出力します。なお，40桁モード（width 40）の場合は，ESCキーを押しても128バイト1ブロックの境目でしか一時停止できません。

アドレスの入力

まず，前回入力したアドレスが表示されます（初期状態は$0000）。リターンキーのみを入力した場合は，その前回入力アドレスが新アドレスとなります。アドレスは16進4桁で入力しますが，エンドアドレスで'.'のみを入力した場合は，エンドアドレスとしてスタートアドレス＋1ブロックの値が採用されます。

オプションの入力（コマンド例1）

コマンド例1の中から，複数コマンドの指定が可能です。

'*'の入力——スタートアドレスからエンドアドレスまでを，連続的に表示します。このオプションなしの場合は，スタートアドレスからエンドアドレスを含むブロックまで，ブロック単位に表示します。

このオプションを使うと必要最小限度しかスクリーンに表示しないため，次のコマンドのためにスクリーン上の出力情報をとっておきたい場合などに有効です。

'／'の入力——画面80モード時でも，1行8バイト1ブロック128バイトの出力形式を指定します。

'＝'の入力——メモリ内容をプリンタに出力します。

リターンキーのみの入力——初期設定のモード，つまりブロック単位に縦横のチェックサム付きで，スクリーン上にダンプします。

ダンプ後の選択コマンド（コマンド例2）

'＋'の入力——現在表示されているメモリブロックの次のブロックを表示します。$FFFFを超えた場合は，$0000に戻ります。

'－'の入力——現在表示されているメモ

リブロックの1つ前のブロックを表示します。$0000を超えた場合は，$FFFFへ移動します。

リターンキーの入力——当ルーチンを終了し，メインコマンドレベルに戻ります。

(2)ライトルーチン

コマンド：W（rite

機能：指定したアドレスから1ブロック表示後，スクリーンエディット風にメモリを修正，書き込みができます。エディットモードとして，16進かASCIIいずれかの形式を選ぶことができ，変更したメモリ内容は，グリーンで表示されます。チェックサムは，1バイト入力ごとに更新され，表示されます。

メモリブロックの指定

そのブロックの先頭アドレスを，16進4桁で入力します。

オプションの入力（コマンド例3）

やはり，複数コマンドの入力が可能です。

'*'の入力——スクロールモードにする。つまり，エディット中にカーソルがエディットブロックから上下にはみ出ようとした場合，それぞれ下上へ1行ずつスクロールします。このモードにすると，ブロックをページ切り換えすることなく，連続的にメモリに書き込むことができます。

'／'の入力——画面80桁モード時，1行8バイト，1ブロック128バイト単位でのエディットを指定します。

リターンキーのみの入力——オプションなし，つまり初期設定のブロック単位（いわばページ切り換えモード）でのエディットとなります。

エディットモードの選択

エディットブロックの表示後，コマンド例4のコマンドラインが表示され，エディットモードの選択を促します。

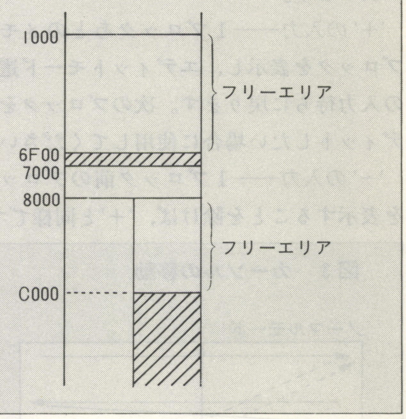
'H'の入力——16進形式のエディットモードに入ります。

入力可能キャラクタは，当然'0'−'F'の16進キャラクタと，カーソル移動のためのキーおよびESC，CTRL-Cキーのみです。なお，'A'—'F'はテンキーから打ち込み可能で，'＊／＋−＝，'がそれぞれ'ABCDEF'に対応します。カーソル移動は，上下左右矢印キーがそれぞれの移動に対応し，リターンキーは次の行の先頭にカーソルを移動させます(コマンド例5参照)。メモリ内容の変更は，カーソルが別のメモリ表示部分に移動した時点で1バイトごとにメモリに落とされ，リアルタイムにグリーンで表示されます。最後にESCキーを押してこのモードから抜け，前掲のエディットモード選択のレベルへ戻ります。

'A'の入力——ASCII形式のエディットモードに入ります。

すべてのキーが入力できます。特殊キー（INS・CLS・DELなど），CTRLコードも，それぞれの対応するコードがそのままメモリに書き込まれます。PFキーも，その定義されている文字列が連続したメモリ領域に書き込まれます。例外は，カーソル移動の上下左右矢印キーとESCキーのみです(コマンド例6)。ESCキーはこのモードを抜

図2　FM-7の場合



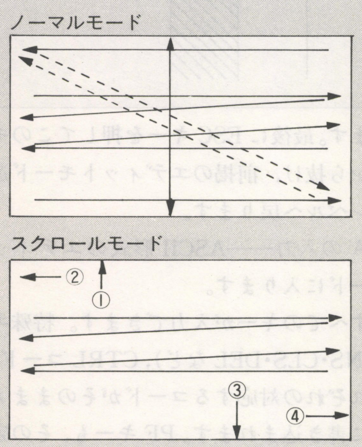

```
コマンド例　1
ｺﾏﾝﾄﾞ: '*':sequential '/':8bytes/line '=':to printer RET:no option
コマンド例　2
ｺﾏﾝﾄﾞ: '+':DOWN '-':UP  RET:QUIT
コマンド例　3
ｺﾏﾝﾄﾞ: '*':scroll mode '/':8bytes/line RET:no option
コマンド例　4
ｺﾏﾝﾄﾞ: 'H':hex 'A':ascii '+':down '-':up RET:quit
コマンド例　5
KEY: '*'=A,'/'=B,'+'=C,'-'=D,'='=E,','=F,'0'-'F':hex RET:lf ARROW:move ESC:end
コマンド例　6
KEY: ARROW:move ESC:end
```

け，前掲のエディットモード選択のレベルに戻ります。

'+'の入力――1ブロックあとのメモリブロックを表示し，エディットモード選択の入力待ちに戻ります。次のブロックをエディットしたい場合に使用してください。

'−'の入力――1ブロック前のブロックを表示することを除けば，'+'と同様です。

図3　カーソルの移動



① 1行下へスクロールしカーソル位置は変化しない
② 1行下へスクロールしカーソル位置は新先頭行の末尾に移動する
③ 1行上へスクロールしカーソル位置は変化しない
④ 1行上へスクロールしカーソル位置は新最終行の先頭へ移動する

リターンキーの入力――当ルーチンを終了し，メインコマンドレベルに戻ります。

（カーソルの移動の仕方については，**図3**を参照）

(3)ブロック転送ルーチン

コマンド：T（ransfer

機能：指定したメモリ範囲の内容を，指定したメモリ位置へ転送します。

ベリファイオプションを選択した場合には，転送前に転送元と転送先のメモリ内容を表示し，確認を求めてきますので，OKならリターンキー，キャンセルならCTRL-Cキーを押してください。OKの場合は，転送後にさらに転送先のメモリ内容を表示します。

(4)メモリフィルルーチン

コマンド：F（ill

機能：指定したメモリ範囲を，指定した1バイトの内容で埋めます。

フィルキャラクタの入力

16進，10進，ASCIIキャラクタの3通りの入力方法があります（コマンド例7参照）。16進の場合は，'0'−'F'を2桁で入力し，10進の場合は'#'と0から255までの数字を入力します。またASCIIの場合は，'"'とASCIIキャラクタを1文字入力します。

(5)カリキュレートルーチン

コマンド：C（alculate

機能：16進数，10進数，2進数それぞれの基数変換と2項四則演算を行い，結果を16進数，10進数，2進数で表示します。アドレス計算，基数変換に便利です。

コマンド例8にあるように，16進数は必ず4桁で入力します。10進数は数字の前に'#'をつけ，0から65535の範囲で入力し，2進数は数字の前に'%'をつけ，8か16ビットの0,1で入力してください。四則演算記号はBASIC同様'＋−＊／'です。式は3項以上指定しても，最初の2項についてのみ演算を行います。また，演算は符号なしの16ビットで行い，結果がこれに収まりきれない場合のあふれ分は無視します。

1項だけを入力した場合は，基数変換となります。リターンのみを入力すると，当ルーチンを終了してメインコマンドレベルに戻ります。

(6)サーチルーチン

コマンド：S（earch

機能：1〜16バイトまでの長さのストリング(ASCIIキャラクタ以外もOK)を，指定メモリ範囲のなかでサーチし，一致した場合にその先頭アドレスを表示する。後述

コマンド例　7

```
VALUE: HEX:2(00-FF) DECIMAL:'#'+(0-255) CHAR:'"'+ASCII CHAR
```

コマンド例　8

```
シキ: HEX:4(0000-FFFF) DECIMAL:'#'+(0-65535) BIN:'%'+(8,16BIT-01) RET:quit
```

コマンド例　9

```
TARGET: HEX:00-FF(1-16) ASCII:'"'+Ascii Char(1-16) RET:default
```

コマンド例　10

```
ｺﾏﾝﾄﾞ: n:find count(1-255,'/'=255) '*':verify RET:count=1,no verify
```

コマンド例　11

```
ｺﾏﾝﾄﾞ: register name:for Change('CC'-'SP') RET:quit
```

コマンド例　12

```
ｺﾏﾝﾄﾞ: address:Hex4(7 points,terminated space) '*':clear brk point RET:quit
```

のオプションにより，サーチ回数の指定とベリファイモードが選べます。すぐエディットモードに入ることも可能なので，特定ストリングの修正などに便利です。

サーチターゲットの入力

まず，'/'で囲まれた前回入力したターゲットが表示されます。ターゲットを変更しない場合は，リターンキーのみを入力してください。

16進で入力の場合は，'0'—'F'の16進キャラクタで16バイト分（32文字で当然偶数）の範囲内で，またASCIIキャラクタで入力の場合は，先頭に'"'をつけ続けてASCII文字を16文字の範囲内で，それぞれ入力します。16バイトを超えた分は無視されます（コマンド例9参照)。

オプションの入力（コマンド例10)

'*'の入力——ベリファイオプション。一致したパターンを発見するごとに，その先頭アドレスから16バイト分を表示します。この状態で即座に前述のライトルーチンと同様のエディットモードを選び，修正することが可能です。その後，再び次のメモリ位置からエンドアドレスまで指定回数だけサーチを続行し，以上を繰り返します。

数字の入力——サーチ回数を1～255までの10進数で入力します。'/'を入力した場合は255の指定となります。

リターンキーのみの入力——ベリファイなし，サーチ回数1となります。

〔注意点〕

a）当システム内をベリファイ付きでサーチした場合，ターゲットとは異なるパターンを一致パターンとして表示する場合がありますが，これはメモリ内容を表示するとき，ワークエリアが同一のため書き換えられたもので，サーチのときに一致したパターンがあったからです。バグではありません。

b）FM-7を後述の裏RAMモードで使用している場合は，サーチエンドアドレスとして$FD00以降を指定しないでください。$FD0Fを読んだ場合はROMモードとなり，当システムが隠れてしまってシステムダウンとなります。

例1

```
>REGISTER

CC A  B  DP X    Y    U    PC   SP
A0 00 00 00 4000 95C3 8000 400B 7BFB
EFHINZVC                        D2 ) スタックの下2バイト
10100000— フラグの状態が一目でわかる  0F
Register ?D
D =0000 ?2330—— A，BはDレジスタとして一括変更可
Register ?

>
```

(7)レジスタ内容表示，変更ルーチン

コマンド：R (egister

機能：レジスタ内容を表示します。CCレジスタは2進ビットでも表示されますので，フラグの状態が一目で判別できます。また，SPの指すメモリ内容の下2バイトも表示され，さらにレジスタ名を入力することにより，その内容を変更することができます（例1参照)。

レジスタ内容は，このシステムの起動時か，後述のブレークポイントでブレークした時点での内容です。

レジスタ内容の変更（コマンド例11)

変更するレジスタ名を入力します。表示に変更がない場合はリターンキーを押し，変更する場合は，8ビットレジスタなら16進2桁で，16ビットレジスタなら16進4桁で入力します。CCレジスタは2進8ビットでも入力可能です。

なお，変更されたレジスタはそのまま，Goルーチンでユーザプログラム，Endルーチンで当システムの呼び出し元へそれぞれ引き渡されるので，各種レジスタの変更は，あくまでも使う側の責任において行ってください。

(8)ゴールーチン

コマンド：G (o

機能：ブレークポイントを設定し，指定アドレスから実行を開始する。ブレークポイントは，後述のブレークポイント設定ルーチンで設定されます。

ブレークするとレジスタ内容と次のブレークポイントを表示し，さらにGoするのか確認を求めてきます（ポイントがまだ残っている場合)。この状態でリターンキーを押すと，前回のポイントから実行を再開します。CTRL-Cを押した場合は，一旦メインコマンドレベルに戻りますので，必要に応じてメモリを見るなり，修正したり，他のコマンドを使用してください。その後，またこのルーチンに入ることにより，引き続き次のブレークポイントへGoできます。

なおこのルーチン内では，1個所ずつしかブレークポイントを設定することができないので，複数ポイントを指定しておきたい場合はBrkコマンドを使います。

(9)ブレークポイント設定ルーチン

コマンド：B (rk

機能：最大7個所までのブレークポイントを設定します。

まず，現在設定されているブレークポイ

ントがすべて表示され，コマンド例12が表示されます。

'*'の入力——現在設定してあるブレークポイントをすべて解除します。

ブレークポイントの入力——16進4桁で入力し，最大7個所設定可能で，ポイント間は1個のスペースで区切って入力します。

リターンキーのみの入力——当ルーチンを終了し，メインコマンドレベルに戻ります。

ブレークをかけられるアドレスは，機械語命令の先頭バイトアドレスです（**例2**参照)。それ以外に設定した場合のユーザプログラムの動きは保証できません。また，当システム自身と，ROM内ルーチンおよびサブシステムをアクセスしているルーチンの内部には，ブレークポイントを設定できません。

⑽エンドルーチン

コマンド：E (nd

機能：当システムを終了し，元へ戻ります。

もし，F-BASICのコマンドレベルから当システムを起動したならば，F-BASICのコマンドレベルへ戻り，また，BASICプログラムからデバッグのためにサブルーチンコールで起動された場合には，そのBASICプログラムへ戻り，次の文から実行を再開します。

具体的使用方法

さまざまな方面に役立つこと請け合いですが，ここでは簡単に3つの例について説明します。

◎使用例1 マシン語ダンプリストの打ち込み

STEP1——F-BASICのコマンドレベルより，CLEARコマンドを実行します。

CLEAR, &H1FFF (このアドレスは，当システムのロードアドレスとダンプリストの打ち込みアドレスを考慮のうえ決定する)

STEP2——当システムをメモリ上にロードします。

LOADM"MONITOR"(必要に応じてオフセットをかける)

STEP3——画面の桁数を選択します。1行8バイト1ブロック128バイト仕様の場合は40桁モード，1行16バイト1ブロック256バイト仕様の場合は80桁モードを選択します。ダンプリストのチェックサムのバイト数に合わせます。もちろん，80桁モードでも128/ブロックに設定可能です。

WIDTH 80, 25

STEP4——当システムを起動する。

EXEC &H4000 (ロードオフセットに合わせてアドレスを変える)

STEP5——メインコマンドレベルより'W'か'w'を入力してライトルーチンに入ります。以降は，前述のライトルーチンの項を参照してください。なお，FM-11の場合は，オプションでスクロールモードにすると，使い勝手が格段と向上します。FM-7，8の場合は，スクロールスピードが遅いため，通常のページモードで使用することをお勧めします。

STEP6——打ち込みが終了したら，このライトルーチンを抜けてメインコマンドレベルより'E'か'e'を入力し，F-BASICのコマンドレベルに戻ります。

STEP7——打ち込んだリストをセーブします。

SAVEM"ファイル名", &H○○○○, &H△△△△, &H××××

◎使用例2 マシン語サブルーチンのデバッキング(BASICからの)

STEP1——BASICプログラムとマシン語サブルーチンをメモリ上にロードします。

LOAD"ファイル名"

LOADM"ファイル名"

STEP2——当システムを上記プログラムとダブらないよう，必要に応じてオフセットをかけてロードします。

LOADM"MONITOR" (, &H△△△△)

STEP3——BASICプログラムのリストをとり，CLEAR文をチェックします。次にマシン語ルーチンを呼び出すEXEC文，もしくはDEFUSR文のアドレスを当システムの入口アドレスに書き換えます。

(例) EXEC &H3A00

——► EXEC &H4000

STEP4——BASICプログラムを走らせます。

RUN

STEP5——マシン語ルーチンをコールする文にぶつかると，当システムに制御が移りますので，必要なコマンドを実行させます。デバッグ対象ルーチンを実行するためには，Goコマンドを使い，ブレークポイントをかけ，実行します。

STEP6——デバッグが完了したら，メインコマンドレベルより'E'か'e'を入力します。ここで，このシステムを呼び出したBASICプログラムに制御が移り，次の文から実行が再開されます。

例2

```
00320  3F00  E6    80          ldb   ,x+
00330  3F02  AE    84          ldx   ,x
00340  3F04  31    8D 00AF     leay  msbuf,pcr
00350  3F08  A6    80     lop  lda   ,x+
00360  3F0A  A7    A0          sta   ,y+
```

└── このアドレスにブレークポイントをかける

◎使用例3　裏RAMの活用（FM-7のみ）

裏RAMを利用することにより，当システムを裏RAMより実行し，広いフリーエリアを充分に活用することができます。

STEP1――BASICのコマンドレベルよりCLEAR文を実行します。

CLEAR, &HØFFF　（フリーエリアの下限を決める）

STEP2――ダンプリスト2の裏RAM利用プログラムと，当システムをメモリ上にロードします。オフセットをかけないでください（つまり，裏RAM利用プログラムを &H6D00に，当システムを &H4000にロードする）。

LOADM"URA"（裏RAM利用プログラム）

LOADM"MONITOR"

STEP3――裏RAM利用プログラムを実行させます。

EXEC &H6DØØØ

メッセージが表示されて，すぐBASICコマンドレベルに戻ります。

STEP4――以上で準備完了です。もしマシン語プログラムをデバッグするのであれば，それをメモリ上のフリーエリアにロードします。このあと，裏RAM管理ルーチン（裏RAM利用プログラムに含まれているので，既にメモリ上にロードされている）を実行します。

EXEC &H6FØØ

これで裏RAM上の当システムへ制御が移ります。この状態でのフリーエリアは図2のとおりとなり，広いRAM空間を利用できます。裏RAM使用上の**注意事項**は，以下のとおりです。

① 裏RAM利用プログラムは$6D00，当システムは$4000の位置に，それぞれ必ずロードしてください（つまり，ダンプリストどおりのアドレスから打ち込んでセーブした場合は，オフセットをかけずにロードする）。

② デバッグするマシン語プログラムの制限事項は，以下のとおりです。

a）BASICインタプリタ内ルーチンを利用しているものは不可

b）BIOSとサブシステムをアクセスしているルーチンはOK（BIOSも裏RAM上に移動してある）

c）BASICプログラムからコールされるルーチンは，a）とb）の条件に合っていれば単独でデバッグ可能

③ マシン語ダンプリストの打ち込みは，広いフリーエリアをフルに使える。ただし，当然のことながら，裏RAMの$8000～$BFFFまでのフリーエリアを使って打ち込んだ場合は，当システムを終了する前にT（ransferコマンドを使って，$6EFFよりも前のフリーエリアに転送しておいてください。さもないとセーブできないことになります。

おわりに

以上で説明を終わります。舌足らずでわかりにくい点が多々ありますが，説明よりも実際に走らせてみた方が，はるかにわかりやすいと思います。ぜひご活用いただいて，貴重なご意見，不明な点などをお聞かせください。改良して完成度を高めていきたいと思います。

なお，次号以降で，このプログラムで使われている汎用性のある基本ルーチンについて，ソース付きで解説していきたいと思います。インプット，プリント，基数変換など，下位レベルのルーチン（マシン語でプログラムを組むうえで，一番やっかいな部分）ですから，きっと役に立つと思います。乞うご期待！

ダンプリスト1　MONITOR（FM-7，8用）

（FM-11は$4096，$409Aを変更してください）

© 1984　Y. TAKAHASHI

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 10 EF 8D 19 A2 32 8D 1A 20 .↘■.「2■.
4008 22 34 FF 10 EF 8D 16 FA F1 "4..↘■.
4010 C6 02 8D 25 8D 3F 17 00 5D ニ.■%■?..
4018 A3 17 00 AF 17 00 CA 81 CB 」..ッ..ハ▁
4020 FF 27 0B 48 30 8D 16 86 D2 .'.H0■.■
4028 EC 86 AD 8B 20 EE 17 12 E1 ●■ュ■ ／..
4030 E9 35 7F 10 EE 8D 19 6F B0 ♥5..／■.o
4038 39 4F ED 8D 16 BB 33 8D 93 9O○■.サ3■
4040 19 9B 30 8D 19 55 86 0C 71 .┘0■.U■.
4048 36 02 A6 E5 A7 80 5C 37 7D 6.ヲ◣ァ_¥7
4050 02 4A 26 F4 39 CC 00 00 6B .J&日97..
4058 17 12 16 86 32 17 12 1C 3C ...■2...
4060 17 11 AE 17 11 8B 8E FC 13 ..ョ..▌■
4068 83 A6 84 81 28 25 0B CC 52 ▃ヲ▄▁(%.7
4070 01 00 ED 8D 16 81 86 50 E8 ..○■.▁■P
4078 20 09 CC 00 80 ED 8D 16 05  .7.▁○■.
-----------------------------------------
SUM  CB 26 3A 7E 83 97 9D B6 16
```

```
                                    B8
              00          0F        BA
ADDR +0 +1 |+2 +3 +4 +5 |+6 +7 =S |01234567
4080 76 86 |28 A7 8D 16 |67 17 EC |v■(ァ■.g.
4088 11 7C |86 00 17 11 |8E 17 E0 |.¦■...■.
4090 11 5F |8E FC 82 86 [0C] A7 [B5] ._■ ▃■.ァ
4098 80 86 [13] A7 84 17 11 61 [CD] ▁■.ァ■..a
40A0 6F 8D 16 5B 6F 8D 16 5A D9 o■.[o■.Z
40A8 6F 8D 16 5D 6F 8D 16 53 D4 o■.]o■.S
40B0 6F 8D 16 7D 33 8D 19 25 8D o■.}3■.%
40B8 17 07 F2 39 30 8D 11 14 2B ..〒90■..
40C0 BF FF F4 30 8D 26 A4 BF F8 ソ.日0■&､ソ
40C8 FF FA 39 17 11 7F 2A 2A 2D . 9...**
40D0 20 46 4D 2D 4D 4F 4E 49 13  FM-MONI
40D8 54 4F 52 20 56 31 2E 30 FA TOR V1.0
40E0 20 2A 2A 0D 0A 0D 0A FF A1  **.....
40E8 39 86 3E 17 11 9F 17 00 DB 9■>..ﾉ..
40F0 CE 17 01 2E EC 8D 15 F7 99 ホ...●■.秒
40F8 17 11 76 17 01 7E 17 00 4B ..v..~..
-----------------------------------------
SUM  EC FB [2E] B5 34 D4 [FF] 74 [45]―35
             1B             02
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4100 E0 39 17 01 B4 17 01 C1 BE =9..I..ﾁ
4108 17 02 6B 17 03 30 17 03 E8 ..k..0..
4110 56 A6 8D 15 ED 27 0D 6F 2E Vｦ■.O'.o
4118 8D 15 E7 EC 8D 15 D4 C3 AE ■.◤●■.ﾔﾃ
4120 00 01 20 08 EC 8D 15 CD 84 .. .●■.ﾍ
4128 E3 8D 15 CB ED 8D 15 C5 A4 ┤■.ﾋO■.ﾅ
4130 25 07 10 A3 8D 15 CD 23 71 %...｣■.ﾍ#
4138 D5 17 03 58 81 00 27 30 1F ﾕ..X▁.'0
4140 81 02 27 1F EC 8D 15 AF 06 ▁.'.●■.ｯ
4148 58 49 ED 8D 15 AB EC 8D 54 XIO■.ｫ●■
4150 15 A3 A3 8D 15 A3 ED 8D 1A .｣｣■.｣O■
4158 15 9B ED 8D 15 A6 17 11 0D .┘O■.ｦ..
4160 B9 20 AB 17 11 B4 EC 8D D9 ｹ ｫ..ｴ●■
4168 15 8B ED 8D 15 96 20 9E 83 .▌O■.│ ╲
4170 17 02 A0 17 11 A4 17 11 AD ....､..
4178 A1 39 17 03 8C 17 03 9A 34 ｡9..■..└
-----------------------------------------
SUM  40 11 31 6B 06 38 42 8B F8
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4180 17 03 AD 17 04 57 17 11 61 ..ｭ..W..
4188 62 17 04 5E 17 02 D8 17 E3 b..^..ﾘ.
4190 04 6C 4D 27 20 81 01 26 AC .lM'  ▁.&
4198 05 17 04 D6 20 E8 81 02 81 ...ﾖ ♠▁.
41A0 26 05 17 04 DE 20 DF 81 A4 &...ﾞ ﾟ▁
41A8 04 22 05 17 04 E6 20 DF 2B ."...◥ ﾟ
41B0 17 05 90 20 DA 17 02 5B 1A ..┴ ﾚ..[
41B8 17 11 5F 17 11 5C 39 17 5B .._..¥9.
41C0 11 5F ED 8D 15 29 C1 16 FF ._O■.)ﾁ.
41C8 23 10 17 11 4D 17 11 4A 1A #...M..J
41D0 EC 8D 15 1B C6 16 ED 8D FF ●■..ﾆ.O■
41D8 15 15 4F C6 17 17 10 91 0E ..Oﾆ...┬
41E0 39 34 02 17 11 3B ED 8D 4C 94...;O■
41E8 15 05 30 8D 15 89 C6 50 8B ..0■.▌ﾆP
41F0 86 07 A7 80 6F 80 5A 26 23 ■.ｧ▁o▁Z&
41F8 F7 86 FF A7 84 A6 8D 14 EE 秒■.ｧ▂ｦ■.
-----------------------------------------
SUM  DA B1 4D 0E 80 92 14 B7 C3
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4200 ED 4A 81 28 23 04 C6 17 E4 OJ▁(#.ﾆ.
4208 20 02 C6 18 1F 02 4F C6 36  .ﾆ...Oﾆ
4210 17 30 8D 15 62 17 11 77 EA .0■.b..w
4218 EC 8D 14 D3 17 10 52 35 0E ●■.ﾓ..R5
4220 02 39 17 06 20 86 04 17 19 .9.. ■..
4228 10 6C 17 10 20 20 44 28 4F .l..  D(
4230 75 6D 70 20 57 28 72 69 CC ump W(ri
4238 74 65 20 54 28 72 61 6E B6 te T(ran
4240 73 66 65 72 20 46 28 69 A7 sfer F(i
4248 6C 6C 20 47 28 6F 20 43 39 ll G(o C
4250 28 61 6C 63 75 6C 61 74 0E (alculat
4258 65 20 53 28 65 61 72 63 9B e S(earc
4260 68 20 52 28 65 67 69 73 AA h R(egis
4268 74 65 72 20 42 28 72 6B B2 ter B(rk
4270 20 45 28 6E 64 FF 86 07 EB  E(nd.■.
4278 17 10 1B 39 17 10 21 81 44 ...9..!▁
-----------------------------------------
SUM  8A AD F1 E5 BE 8D 30 88 10
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4280 00 27 F9 C1 45 27 04 C1 12 .' ﾁE'.ﾁ
4288 65 26 08 1F 98 17 0F FD 6D e&..┌..
4290 86 FF 39 4F 30 8D 14 36 14 ■.9O0■.6
4298 E1 86 27 0B 4C 34 02 A6 C1 ├■'.L4.ｦ
42A0 86 27 0F 35 02 20 F1 44 48 ■'.5. 円D
42A8 34 02 1F 98 17 0F DE 35 26 4..┌..ﾞ5
42B0 02 39 35 02 17 10 0A 20 C3 .95.....
42B8 C3 17 0F 91 0D 0A 3E 44 13 ﾃ..┬..>D
42C0 55 4D 50 0D 0A 0D 0A FF 1F UMP.....
42C8 39 17 0F 81 20 53 74 61 28 9..▁ Sta
42D0 72 74 20 41 64 64 72 20 A1 rt Addr
42D8 FF 17 05 26 25 EB 17 0F 77 ...&%♣..
42E0 6C 20 45 6E 64 20 20 20 03 l End
42E8 41 64 64 72 20 FF EC 8D 13 Addr .●■
42F0 14 01 31 8D 14 7C 17 11 8B ..1■.¦..
42F8 1E 17 0F CA 17 0F 4E 20 A2 ...ﾊ..N
-----------------------------------------
SUM  29 D6 40 C6 F8 A1 B8 E4 3A
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4300 3F FF 17 11 9A 30 8D 14 D1 ?...└0■.
4308 29 A6 84 81 FF 26 04 4F 4C )ｦ▂▁.&.O
4310 17 0A 58 A6 84 81 00 27 4B ..Xｦ▂▁.'
4318 3B A6 01 81 2E 26 1C 86 59 ;ｦ.▁.&.■
4320 01 A7 8D 13 DE EC 8D 13 B2 .ｧ■.ﾞ●■.
4328 C8 E3 8D 13 CA 24 03 CC 08 ﾈ┤■.ﾊ$.ﾌ
4330 00 00 83 00 01 ED 8D 13 11 ..▃..O■.
4338 BA 20 19 A6 80 81 04 26 C4 ｺ .ｦ▁▁.&
4340 0E 17 11 01 25 09 17 11 8D ....%...
4348 24 ED 8D 13 A6 20 05 17 93 $O■.ｦ ..
4350 05 03 20 8A EC 8D 13 9B D9 .. ▌●■.┘
4358 10 A3 8D 13 94 24 06 17 28 .｣■.▔$..
4360 04 F3 16 FF 64 EC 8D 13 FC .月..d●■.
4368 88 ED 8D 13 88 EC 8D 13 29 ▎O■.▎●■.
4370 82 ED 8D 13 8F 39 17 07 F5 ▂O■.┼9..
4378 14 17 FE 43 17 06 BD EC 32 .. C..ｽ●
-----------------------------------------
SUM  A6 8D 23 9E 51 6C F1 1B BD
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4380 8D 13 6C 17 0E EB 17 11 44 ■.l..♣..
4388 16 30 8D 13 A5 A6 84 81 36 .0■.･ｦ▂▁
4390 FF 26 05 86 01 17 09 D3 A4 .&.■...ﾓ
4398 A6 80 26 02 20 5E 34 02 02 ｦ▁&. ^4.
43A0 C6 2A 17 11 94 81 00 27 54 ﾆ*..▔▁.'
43A8 0A 86 01 A7 8D 13 53 A7 D2 .■.ｧ■.Sｧ
43B0 8D 13 59 A6 8D 13 37 81 F7 ■.Yｦ■.7▁
43B8 28 26 07 86 01 B7 57 0C F6 (&.■.ｷW.
43C0 20 29 35 02 34 02 C6 2F AB  )5.4.ﾆ/
43C8 17 11 6E 81 00 27 1C 17 71 ..n▁.'..
43D0 06 DF A6 8D 13 2D 27 13 92 .ﾟｦ■.-'.
43D8 EC 8D 13 15 C3 00 7F ED D0 ●■..ﾃ..O
43E0 8D 13 10 ED 8D 13 1D 6F C9 ■..O■..o
43E8 8D 13 18 35 02 C6 3D 17 09 ■..5.ﾆ=.
43F0 11 47 81 00 27 0E 86 01 95 .G▁.'.■.
43F8 A7 8D 13 03 6F 8D 13 0C 65 ｧ■..o■..
-----------------------------------------
SUM  C8 72 B4 E0 B2 2E 34 9B 7D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4400 17 FD DE 39 A6 8D 13 04 75 . ﾞ9ｦ■..
4408 26 F2 17 FD D4 17 06 8B A8 &年. ﾔ..▌
4410 16 FF 63 A6 8D 12 F2 27 D6 ..cｦ■.年'
4418 10 EC 8D 12 EA ED 8D 12 11 .●■.◆O■.
4420 D6 A6 8D 12 E1 A7 8D 12 42 ﾖｦ■.├ｧ■.
4428 D5 6F 8D 12 D2 6F 8D 12 C3 ﾕo■.ﾒo■.
4430 D8 6F 8D 12 CE 6F 8D 12 C2 ﾘo■.ﾎo■.
4438 D3 6F 8D 12 D5 39 EC 8D 68 ﾓo■.ﾕ9●■
4440 12 AF 31 8D 13 FB 17 0F B3 .ｯ1■. ..
4448 CE EC 8D 12 A6 31 8D 13 D0 ﾎ●■.ｦ1■.
4450 F5 17 0F C3 31 8D 13 DF 8E 時..ﾃ1■.ﾟ
4458 A6 8D 12 A3 27 05 17 0E 39 ｦ■.｣'...
4460 8F 20 03 17 0E 60 39 AE 1E ┼ ...`9ｮ
4468 8D 12 8A 17 05 64 17 03 C3 ■.▌..d..
4470 F7 17 03 1D EC 8D 12 84 3D 秒...●■.▂
4478 83 00 01 ED 8D 12 7D 26 B3 ▃..O■.}&
-----------------------------------------
SUM  CA 55 89 73 E4 82 D8 F5 4E
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4480 F0 A6 8D 12 7D 26 09 17 F8 ×ｦ■.}&..
4488 05 76 17 04 D8 17 04 81 0A .v..ﾘ..▁
4490 17 04 7E 39 17 05 97 17 9C ..~9..▕.
4498 FD 25 17 03 A8 86 04 17 85 %..ｨ■..
44A0 0D F4 17 0D A8 20 27 2B 3F .日..ｨ '+
44A8 27 3A 44 4F 57 4E 20 20 D9 ':DOWN
44B0 27 2D 27 3A 55 50 20 20 9A '-':UP
44B8 52 45 54 3A 51 55 49 54 68 RET:QUIT
44C0 20 20 3F FF 86 07 17 0D 2F   ?.■...
44C8 CD EC 8D 12 22 17 0D A1 3F ﾍ●■."..｡
44D0 17 0D CD 81 00 27 F9 C1 53 ..ﾍ▁.' ﾁ
44D8 03 26 05 86 01 17 08 8B 5F .&.■...▌
44E0 C1 2B 26 04 86 02 20 06 C4 ﾁ+&.■. .
44E8 C1 2D 26 0B 86 01 34 02 DC ﾁ-&.■.4.
44F0 1F 98 17 0D 98 20 07 C1 5B .┌..┌ .ﾁ
44F8 0D 26 09 4F 34 02 17 FC D4 .&.O4..
-----------------------------------------
SUM  6B 3A 19 A5 3A 5C EF 44 2C
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4500 E0 35 02 39 17 0D BA 20 4E =5.9..ｺ
4508 C7 17 0D 41 0D 0A 3E 57 D8 ｽ..A..>W
4510 52 49 54 45 0D 0A 0D 0A 62 RITE....
4518 FF 39 17 0D 30 20 57 72 75 .9..0 Wr
4520 69 74 65 20 41 64 64 72 DD ite Addr
4528 20 FF 17 02 D5 25 EB 39 56  ...ｭ%♣9
4530 17 05 5A 17 FC 89 17 03 2C ..Z. ▌..
4538 0C 86 04 17 0D 58 17 0D 36 .■...X..
4540 0C 20 27 2A 27 3A 73 63 B4 . ´*´:sc
4548 72 6F 6C 6C 20 6D 6F 64 19 roll mod
4550 65 20 20 27 2F 27 3A 38 94 e  ´/´:8
4558 62 79 74 65 73 2F 6C 69 2B bytes/li
4560 6E 65 20 20 52 45 54 3A 38 ne  RET:
4568 6E 6F 20 6F 70 74 69 6F 28 no optio
4570 6E FF 86 07 17 0D 1F EC 29 n.■....●
4578 8D 11 74 17 0C F3 17 0F 4E ■.t..月..
-----------------------------------------
SUM  C0 D8 B5 EB 4E 61 54 BA F5
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4580 1E 30 8D 11 AD A6 80 81 40 .0■.ｭｦ__
4588 FF 26 05 86 01 17 07 DB AA .&.■...ﾛ
4590 4D 26 0A 6F 8D 11 7B 6F 74 M&.o■.{o
4598 8D 11 6E 20 29 34 02 C6 51 ■.n )4.ﾆ
45A0 2A 17 0F 95 4D 27 0A 86 E9 *..┤M´.■
45A8 01 A7 8D 11 65 A7 8D 11 F0 .ｧ■.eｧ■.
45B0 5B 35 02 C6 2F 17 0F 81 2E [5.ﾆ/..._
45B8 4D 27 13 A6 8D 11 2F 81 7B M´.ｦ■./_
45C0 28 27 03 17 04 EB 6F 8D 54 (´...♣o■
45C8 11 42 17 FC 14 39 6D 8D AD .B. .9m■
45D0 11 3A 26 F2 17 FC 0A 17 97 .:&年. ..
45D8 04 C1 16 FF 53 EC 8D 11 B7 .ﾁ..S●■.
45E0 10 ED 8D 11 10 ED 8D 11 36 .○■..○■.
45E8 1B 39 17 0C 60 5B 4D 45 C4 .9..`[ME
45F0 4D 4F 52 59 20 45 44 49 39 MORY EDI
45F8 54 5D 0D 0A FF 39 17 04 1B T]...9..
-----------------------------------------
SUM  E4 DD 14 BC E3 CA 81 0F CE
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4600 2D 17 FB BB 17 02 3E 86 D7 -.ｻ..>■
4608 04 17 0C 8A 17 0C 3E 20 32 ...▌..>
4610 27 48 27 3A 68 65 78 20 35 ´H´:hex
4618 27 41 27 3A 61 73 63 69 69 ´A´:asci
4620 69 20 27 2B 27 3A 64 6F 0F i ´+´:do
4628 77 6E 20 27 2D 27 3A 75 2F wn ´-´:u
4630 70 20 52 45 54 3A 71 75 9B p RET:qu
4638 69 74 FF 86 07 17 0C 56 E2 it.■...V
4640 EC 8D 10 AB 17 0C 2A 17 98 ●■.ｫ..*.
4648 0C 56 4D 27 FA C1 03 26 BA .VM´ ﾁ.&
4650 05 86 01 17 07 15 C1 0D 8D .■....ﾁ.
4658 26 03 4F 20 11 30 8D 12 78 &.O .0■.
4660 82 86 06 17 0E D3 4D 26 79 _■...ﾓM&
4668 05 17 0C 55 20 D9 17 FB 88 ...U ﾙ.
4670 70 39 EC 8D 10 7F E3 8D 21 p9●■..ﾖ■
4678 10 7D ED 8D 10 77 ED 8D 08 .}○■.w○■
-----------------------------------------
SUM  62 98 85 65 1D 4C 21 75 E3
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4680 10 82 39 EC 8D 10 6E A3 65 ._9●■.n｣
4688 8D 10 6C ED 8D 10 66 ED E6 ■.l○■.f○
4690 8D 10 71 39 4F 17 04 CD 7E ■.q9O..^
4698 CC 05 03 17 0B D3 4F 17 2F ﾌ....ﾓO.
46A0 04 61 6F 8D 10 6B 17 0B FE .ao■.k..
46A8 F7 4D 27 FA C1 03 26 0B 5A 秒M´ ﾁ.&.
46B0 CC 00 15 17 0B BB 86 01 45 ﾌ....ｻ■.
46B8 17 06 B0 C1 1C 27 46 C1 D8 ..ｰﾁ.´Fﾁ
46C0 1D 27 47 C1 1E 27 48 C1 9A .´Gﾁ.´Hﾁ
46C8 1F 27 49 C1 1B 27 6A C1 BD .´Iﾁ.´jﾁ
46D0 0D 27 46 C1 30 25 06 C1 57 .´Fﾁ0%.ﾁ
46D8 39 22 02 20 23 30 8D 12 6F 9". #0■.
46E0 08 86 12 17 0E 53 4D 27 8C .■...SM´
46E8 35 80 01 34 02 CC 00 03 BB 5_.4.ﾌ..
46F0 36 06 35 04 4F 36 06 17 17 6.5.O6..
46F8 0E 0E 37 06 37 10 CB 41 AC ..7.7.ﾋA
-----------------------------------------
SUM  D7 0C CB 40 8E 62 93 23 94
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4700 17 04 F2 20 A1 17 05 32 1C ..年 ｡..2
4708 20 9C 17 05 4B 20 97 17 F1  ┌..K ┃.
4710 05 64 20 92 17 05 73 20 CA .d ┤..s
4718 8D 17 05 82 20 88 17 06 F0 ■.._ ▌..
4720 72 5D 27 03 17 0B 61 86 02 r]´...a■
4728 07 17 0B 6A 17 08 59 6F 7A ...j..Yo
4730 8D 0F DE 17 0B 8B 16 FF 3C ■.ﾞ..▌..
4738 6D 4F C6 16 17 0B 32 17 03 mOﾆ...2.
4740 FA 9F 39 86 01 17 04 1D 91  ┘9■....
4748 86 01 17 03 B6 EC 8D 0F DF ■...ｶ●■.
4750 C2 17 0B 1D 17 0B 49 4D B9 ﾂ.....IM
4758 27 FA C1 1C 27 15 C1 1D 18 ´ ﾁ.´.ﾁ.
4760 27 16 C1 1E 27 17 C1 1F 3A ´.ﾁ.´.ﾁ.
4768 27 18 C1 1B 27 19 17 05 77 ´.ﾁ.´...
4770 49 20 E1 17 05 8C 20 DC EE I ┣..▌ ﾜ
4778 17 05 97 20 D7 17 05 A2 68 .. ┃ﾗ..｢
-----------------------------------------
SUM  53 F1 1A 05 92 63 C0 B2 CA
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4780 20 D2 17 05 A1 20 CD 4F EB  ﾒ..｡ ﾍO
4788 C6 16 17 0A E4 17 FA 51 43 ﾆ...◢. Q
4790 39 A6 8D 0F 69 A7 8D 0F 27 9ｦ■.iｧ■.
4798 67 33 8D 12 40 31 8D 0F 46 g3■.@1■.
47A0 D6 1F 10 17 01 18 A6 80 5B ﾖ.....ｦ_
47A8 36 02 17 0C 7D ED A1 86 EC 6...}○｡■
47B0 20 A7 A0 6A 8D 0F 49 26 DC  ｧ.j■.I&
47B8 ED 86 FF A7 A4 31 8D 0F 8A ○■.ｧ､1■.
47C0 B6 17 01 09 17 01 59 31 79 ｶ.....Y1
47C8 8D 0F AC 33 8D 12 0E A6 CE ■.ｬ3■..ｦ
47D0 8D 0F 2B A7 8D 0F 29 A6 D9 ■.+ｧ■.)ｦ
47D8 C2 17 01 04 A7 A0 A6 8D 58 ﾂ...ｧ.ｦ■
47E0 0F 1E 81 09 26 04 86 20 87 .._.&.■
47E8 A7 A0 6A 8D 0F 12 26 E7 6C ｧ.j■..&◤
47F0 86 FF A7 A0 31 8D 0F 7F 18 ■.ｧ.1■..
47F8 17 00 D2 17 00 F6 17 01 0E ..ﾒ..分..
-----------------------------------------
SUM  84 18 4B 98 1B AF 06 8A D9
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4800 BD 39 EC 8D 0E EB 31 8D 26 ｽ9●■.♣1■
4808 0F 68 17 0C 0A 17 0A B6 7B .h.....ｶ
4810 17 0A 3A 20 3F FF 17 0C DC ..: ?...
4818 86 30 8D 0F 15 A6 84 81 12 ■0■..ｦ▄_
4820 FF 26 04 4F 17 05 44 A6 7E .&.O..Dｦ
4828 80 26 02 20 10 81 04 26 83 _&. ._.&
4830 0F 17 0C 11 25 0A 17 0C 95 ....%...
4838 34 ED 8D 0E B4 1C FE 39 C3 4○■.ｴ. 9
4840 8D 13 1A 01 39 86 0C 17 9D ■...9■..
4848 0A 4C 17 0A 00 BA CF DD DD .L...ｺﾏﾝ
4850 C4 DE 3A FF 39 17 09 F5 29 ﾄﾞ:.9..円
4858 20 69 6E 70 75 74 20 65 D5  input e
4860 72 72 6F 72 0D 0A FF 39 14 rror...9
4868 34 10 A6 8D 0E 80 81 28 AE 4.ｦ■.__(
4870 27 0C A6 8D 0E 93 26 06 33 ´.ｦ■.┣&.
4878 31 8D 0F FD 20 04 31 8D AC 1■. .1■
-----------------------------------------
SUM  A4 EC 0C 59 9C 3F 0E 23 01
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4880 0F CC A6 8D 0E 79 27 16 D2 .ﾌｦ■.y´.
4888 17 0A 65 A6 8D 0E 5F 81 A7 ..eｦ■._▁
4890 50 26 06 A6 8D 0E 72 27 56 P&.ｦ■.r´
4898 11 17 0A 77 20 0C 17 0A F6 ...w ...
48A0 25 A6 8D 0E 64 27 03 17 0B %ｦ■.d´..
48A8 0A 70 35 10 39 EC 8D 0E 7F .p5.9●■.
48B0 46 44 56 44 56 44 56 44 58 FDVDVDVD
48B8 56 E7 8D 0E 41 39 17 0B 74 V◤■.A9..
48C0 56 1F 20 C3 00 04 1F 02 7D V. ﾃ....
48C8 86 20 A7 A0 39 34 10 A6 10 ■ ｧ.94.ｦ
48D0 8D 0E 2C 27 05 17 0A 18 2C ■.,´....
48D8 20 03 17 09 E9 35 10 39 AA  ...♥5.9
48E0 81 20 25 0D 81 7F 27 09 03 _ %._.´.
48E8 81 A0 27 05 81 FF 27 01 F5 _.´._.´.
48F0 39 86 2E 39 34 10 A6 8D 9D 9■.94.ｦ■
48F8 0E 05 26 0F A6 8D 0D EE 76 ..&.ｦ■./
-----------------------------------------
SUM  24 EF 6A AD 7F D0 56 BA 89
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4900 81 28 26 02 20 08 17 0A 1A _(&. ...
4908 11 20 03 17 0A 05 35 10 9F .....5.
4910 39 A6 8D 0D EA 26 05 17 A5 9ヲ■.◆&..
4918 0A 00 20 03 17 09 F4 39 7A .. ...日9
4920 34 10 E6 8D 0D D8 4F ED D8 4.◥■.リOO
4928 8D 0D CE 1F 10 A3 8D 0D D4 ■.ホ..」■.
4930 C8 1F 02 A6 8D 0D C7 17 07 ネ..ヲ■.ヌ.
4938 03 F0 34 02 86 05 17 09 D4 .×4.■...
4940 55 35 02 17 0A E4 ED 8D 0B U5...◢O■
4948 0E 28 86 20 A7 8D 0E 24 42 .(■ ァ■.$
4950 86 FF A7 8D 0E 1F 31 8D A4 ■.ァ■..1■
4958 0E 18 17 FF 70 86 07 17 50 ....p■..
4960 09 34 35 10 39 34 10 33 32 .45.94.3
4968 8D 10 72 31 8D 0F 58 17 4B ■.r1■.X.
4970 FF 5B 31 8D 0E 01 AE 8D 62 .[1■..ョ■
4978 0D 7B E6 8D 0D 80 34 04 C0 .{◥■._4.
-----------------------------------------
SUM  FA A8 C4 9B 6B A3 7C B4 3F
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4980 34 10 17 03 B9 36 02 17 66 4...ケ6..
4988 0A A0 ED A1 86 20 A7 A0 25 ..O。■ ァ.
4990 35 10 1F 10 C3 00 01 1F 57 5...テ...
4998 01 35 04 5A 26 E0 17 03 B4 .5.Z&=..
49A0 DB 17 0A 86 ED A1 86 FF 95 ロ..■O。■.
49A8 A7 A4 86 05 17 08 E7 31 0D ァ、■...◤1
49B0 8D 0D C4 17 FF 17 86 07 18 ■.ト...■.
49B8 17 08 DB 35 10 39 34 10 BC ..ロ5.94.
49C0 17 08 DD 81 00 27 08 C1 6D ..ン_.´.チ
49C8 03 26 04 4F 17 03 9C 35 67 .&.O..◜5
49D0 10 39 A6 8D 0D 2C 26 08 E3 .9ヲ■.,&.
49D8 CC 00 10 ED 8D 0D 1D 39 B9 7..O■..9
49E0 E6 8D 0D 1A 4F 36 06 EC 11 ◥■..O6.●
49E8 8D 0D 08 A3 8D 0D 02 36 17 ■..」■..6
49F0 06 17 0B 14 37 06 C3 00 3C ....7.テ.
49F8 01 ED 8D 0C FF 37 06 39 FC .O■..7.9
-----------------------------------------
SUM  0A CA 9A 0C FE 12 A0 B2 DC
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4A00 E6 8D 0C EA C1 50 26 0C AC ◥■.◆チP&.
4A08 A6 8D 0C FD 27 04 C6 28 55 ヲ■. ´.ニ(
4A10 20 02 C6 49 31 8D 0D 5F 5B  .ニI1■._
4A18 86 2D A7 A0 5A 26 FB 86 FB ■-ァ.Z& ■
4A20 FF A7 A4 31 8D 0D 50 17 7C .ァ、1■.P.
4A28 FE A3 17 FE C7 39 17 08 D5 」. ヌ9..
4A30 1C 20 53 65 6C 65 63 74 9C . Select
4A38 20 3F FF 39 17 FE 06 86 38  ?.9. .■
4A40 04 17 08 52 17 08 06 20 BA ...R...
4A48 27 2A 27 3A 73 65 71 75 70 ´*´:sequ
4A50 65 6E 74 69 61 6C 20 20 BD ential
4A58 27 2F 27 3A 38 62 79 74 3E ´/´:8byt
4A60 65 73 2F 6C 69 6E 65 20 CF es/line
4A68 20 27 3D 27 3A 74 6F 20 E8  ´=´:to
4A70 70 72 69 6E 74 65 72 20 24 printer
4A78 20 52 45 54 3A 6E 6F 20 42  RET:no
-----------------------------------------
SUM  37 2E 76 21 BE A0 89 DB BE
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4A80 6F 70 74 69 6F 6E FF 86 1E option.■
4A88 07 17 08 0A 39 17 07 BD 44 ....9..ス
4A90 20 4F 70 74 69 6F 6E 20 B9  Option
4A98 3F FF 39 17 07 AF 20 69 CD ?.9..ッ i
4AA0 6C 6C 65 67 61 6C 20 6F 00 llegal o
4AA8 70 74 69 6F 6E 0D 0A FF 40 ption...
4AB0 39 86 01 A7 8D 0C 52 A7 F9 9■.ァ■.Rァ
4AB8 8D 0C 51 EC 8D 0C 38 ED 94 ■.Q●■.8O
4AC0 8D 0C 44 CC 00 80 ED 8D A3 ■.D7._O■
4AC8 0C 2D A6 8D 0C 30 A7 8D DC .-ヲ■.0ァ■
4AD0 0C 34 86 08 A7 8D 0C 26 34 .4■.ァ■.&
4AD8 39 17 07 C4 4D 27 FA C1 4A 9..トM´ チ
4AE0 03 26 04 4F 17 02 84 C1 DA .&.O..▄チ
4AE8 59 26 02 20 04 C1 79 26 05 Y&. .チy&
4AF0 09 1F 98 17 07 97 86 01 FC ..r.. ■.
4AF8 20 01 4F 34 02 17 08 1A DF  .O4...
-----------------------------------------
SUM  DA 37 A9 46 25 09 6D D1 6C
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4B00 35 02 39 34 02 86 03 A7 D6 5.94.■.ァ
4B08 8D 0C 04 A7 8D 0C 05 86 68 ■..ァ■..■
4B10 12 A7 8D 0B FB 35 02 4D D0 .ァ■. 5.M
4B18 26 21 A6 8D 0B E0 81 08 EE &!ヲ■.=_.
4B20 22 08 86 1B A7 8D 0B E6 F0 ".■.ァ■.◥
4B28 20 06 86 33 A7 8D 0B DE FC  .■3ァ■.゛
4B30 86 05 A7 8D 0B D7 A7 8D D5 ■.ァ■.ラァ■
4B38 0B D9 39 A6 8D 0B BF 81 9B .ル9ヲ■.ソ_
4B40 08 22 11 86 20 A7 8D 0B 20 .".■ ァ■.
4B48 C4 A7 8D 0B C6 86 27 A7 1D トァ■.ニ■´ァ
4B50 8D 0B BB 39 86 38 A7 8D 7E ■.サ9■8ァ■
4B58 0B B3 A7 8D 0B B5 86 48 80 .ウァ■.オ■H
4B60 A7 8D 0B AA 39 34 02 17 6F ァ■.ェ94..
4B68 F6 77 CC 00 16 17 07 01 6E 介w7.....
4B70 17 06 DA 20 20 20 20 20 97 ..レ
4B78 20 20 20 20 20 20 20 FF DF        .
-----------------------------------------
SUM  05 73 2D 35 81 48 31 12 E6
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4B80 CC 00 17 17 06 EB 86 0C 7D 7....♣■.
4B88 17 07 0B 17 06 BF 4B 45 95 .....ソKE
4B90 59 3A FF 86 04 17 06 FE 37 Y:.■...
4B98 35 02 4D 26 3B 17 06 AD AF 5.M&;..ı
4BA0 20 27 2A 27 3D 41 2C 27 69  ´*´=A,´
4BA8 2F 27 3D 42 2C 27 2B 27 7A /´=B,´+´
4BB0 3D 43 2C 27 2D 27 3D 44 A8 =C,´-´=D
4BB8 2C 27 3D 27 3D 45 2C 27 8C ,´=´=E,´
4BC0 2C 27 3D 46 2C 27 30 27 80 ,´=F,´0´
4BC8 2D 27 46 27 3A 68 65 78 40 -´F´:hex
4BD0 20 52 45 54 3A 6C 66 FF 16  RET:lf.
4BD8 17 06 72 20 41 52 52 4F E3 ..r ARRO
4BE0 57 3A 6D 6F 76 65 20 45 AD W:move E
4BE8 53 43 3A 65 6E 64 FF 86 8C SC:end.■
4BF0 07 17 06 A2 39 34 04 17 4E ...「94..
4BF8 01 99 5D 26 20 86 04 17 DE .┐]& ■..
-----------------------------------------
SUM  6B CE 82 0E 3C 7C 11 9B 2D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4C00 06 94 35 04 E7 8D 0B 0D 5F .‾5.◤■..
4C08 1F 98 17 06 80 86 07 17 F8 .r.._■..
4C10 06 84 86 01 A7 8D 0A F9 48 .▄■.ァ■.
4C18 6C 8D 0A F7 39 35 04 17 83 l■.※95..
4C20 01 8C 17 06 63 86 04 17 AE .■..c■..
4C28 06 6C 17 01 B4 6C 8D 0A 41 .l..エl■.
4C30 E2 17 01 F9 86 07 17 06 9D ‡.. ■...
4C38 5D 39 17 01 56 5D 27 14 9C ]9..V]´.
4C40 6D 8D 0A CD 27 0A 6F 8D FE m■.ヘ´.o■
4C48 0A C7 17 01 C3 17 01 91 55 .ヌ..テ..┬
4C50 6C 8D 0A BF 17 01 D6 39 E9 l■.ソ..ヨ9
4C58 17 01 38 5D 26 14 6D 8D E1 ..8]&.m■
4C60 0A AF 27 0A 6F 8D 0A A9 99 .ッ´.o■.ゥ
4C68 17 01 A5 17 01 73 6A 8D 3F ..・..sj■
4C70 0A A1 17 01 FF 39 6D 8D F5 .。...9m■
4C78 0A 97 27 0A 6F 8D 0A 91 69 . ▕´.o■.┬
-----------------------------------------
SUM  0C 4F 8F 19 3F 27 8D A7 9D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4C80 17 01 8D 17 01 5B 17 02 31 ..■..[..
4C88 32 39 6D 8D 0A 83 27 0A 23 29m■.▄´.
4C90 6F 8D 0A 7D 17 01 79 17 2B o■.}..y.
4C98 01 47 17 02 4F 39 6D 8D E3 .G..O9m■
4CA0 0A 6F 27 0A 6F 8D 0A 69 19 .o´.o■.i
4CA8 17 01 65 17 01 33 A6 8D FB ..e..3ヲ■
4CB0 0A 5B A7 8D 0A 5D 17 02 19 .[ァ■.]..
4CB8 33 39 34 04 86 01 17 02 44 394.■...
4CC0 79 35 04 34 04 E7 84 86 DB y5.4.◤▄■
4CC8 04 17 05 CA 86 01 17 03 8B ...ハ■...
4CD0 05 86 07 17 05 C0 17 02 87 .■...タ..
4CD8 44 86 04 17 05 B8 35 02 D9 D■...ク5.
4CE0 17 04 DC 86 07 17 05 AE 4E ..ワ■...ョ
4CE8 86 01 17 02 F5 86 07 17 39 ■...■■..
4CF0 05 A4 A6 8D 0A 1D 81 3F C3 .、ヲ■.._?
4CF8 26 04 6C 8D 0A 15 17 01 5A &.l■....
-----------------------------------------
SUM  A5 17 9B A3 15 65 8D 3C 3D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4D00 2C 39 A6 8D 0A 0D 81 3F 6F ,9ｦ■.._?
4D08 26 04 6C 8D 0A 05 17 01 4A &.l■....
4D10 1C 39 A6 8D 09 FD 81 41 50 .9ｦ■. _A
4D18 26 04 6A 8D 09 F5 17 01 37 &.j■.■..
4D20 53 39 17 01 96 39 17 01 8B S9..|9..
4D28 C3 39 6F 8D 09 D3 E6 A0 5A ﾃ9o■.ﾓ◥.
4D30 EB 8D 09 CD E7 8D 09 C9 94 ■■.ﾍ◤■.ﾉ
4D38 4A 26 F3 1F 98 39 C6 10 29 J&月.┌9ﾆ.
4D40 6F 8D 09 BD 34 04 A6 84 24 o■.ｽ4.ｦ▄
4D48 AB 8D 09 B5 A7 8D 09 B1 E4 ｫ■.ｵｧ■.ｱ
4D50 E6 8D 09 AA 1D ED 8D 09 C6 ◥.ｪ.○■.
4D58 A0 1F 10 E3 8D 09 9A 1F 01 ...╡■.└.
4D60 01 35 04 5A 26 DE A6 8D CB .5.Z&ﾞｦ■
4D68 09 97 39 4D 27 03 17 F4 5B . ■M´..日
4D70 70 10 EE 8D 09 94 33 8D 58 p.◢■.‾3■
4D78 0C 63 10 3F E6 8D 09 7E B8 .c.?◥.~
-----------------------------------------
SUM  05 A4 0A 20 05 5F CB E5 E7
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4D80 5A 34 04 37 02 AB C4 37 71 Z4.7.ｫﾄ7
4D88 04 36 02 35 04 5A 26 F1 E6 .6.5.Z&円
4D90 37 02 39 CC 00 03 36 06 7D 7.97..6.
4D98 E6 8D 09 77 4F 83 00 05 CA ◥.wO▃..
4DA0 36 06 17 07 63 37 06 34 2E 6...c7.4
4DA8 04 37 06 35 02 39 6D 8D AB .7.5.9m■
4DB0 09 5F 27 16 6F 8D 09 59 03 ._´.o■.Y
4DB8 A6 8D 09 59 17 06 C1 34 A7 ｦ■.Y..ﾁ4
4DC0 02 4F 17 01 75 35 02 A7 BC .O..u5.ｧ
4DC8 84 39 4F 17 06 B2 34 02 11 ▄9O..ｲ4.
4DD0 4F 17 01 66 E6 84 C4 F0 EB O..f◥▄ﾄ╳
4DD8 E7 84 35 02 AA 84 A7 84 FB ◤▄5.ｪ▄ｧ▄
4DE0 39 86 04 17 04 B0 17 01 A6 9■..ｰ..
4DE8 9F 17 01 CD E6 8D 09 24 24 ╯..ﾍ◥.$
4DF0 17 04 7E 86 07 17 04 9E DF ..~■...╰
4DF8 86 04 17 04 99 4F 17 01 A5 ■...┐O..
-----------------------------------------
SUM  95 EA CB 48 D5 20 39 62 22
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4E00 D5 86 07 17 04 90 4F 17 73 ﾕ■...┼O.
4E08 01 D8 86 07 17 04 87 39 41 .ﾘ■...■9
4E10 E6 8D 09 01 4F 17 06 68 51 ◥..O..h
4E18 48 48 48 48 34 02 4F 17 BC HHHH4.O.
4E20 01 18 A6 84 84 0F AA E4 64 ..ｦ▄▄.ｪ◢
4E28 A7 84 35 02 39 6C 8D 08 9C ｧ▄5.9l■.
4E30 E2 A6 8D 08 DE A1 8D 08 31 ╪ｦ■.ﾞ｡■.
4E38 D5 23 31 A6 8D 08 CE A7 D9 ﾕ#1ｦ■.ﾎｧ
4E40 8D 08 D0 6C 8D 08 CD A6 D9 ■.ﾐl■.ﾍｦ
4E48 8D 08 C9 A1 8D 08 C1 23 78 ■.ﾉ｡■.ﾁ#
4E50 1B 6D 8D 08 BD 27 0D 17 25 .m■.ｽ´..
4E58 02 1E A6 8D 08 B2 A7 8D 41 ..ｦ■.ｲｧ■
4E60 08 B2 20 08 A6 8D 08 A7 C4 .ｲ .ｦ■.ｧ
4E68 A7 8D 08 A8 EC 8D 08 A3 08 ｧ■.ｨ●■.｣
4E70 17 03 FE 39 6A 8D 08 9B EB .. 9j■.┘
4E78 A6 8D 08 97 A1 8D 08 8D 95 ｦ■. ■｡■.■
-----------------------------------------
SUM  06 02 71 BD 42 EE 1F 49 CE
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4E80 24 31 A6 8D 08 88 A7 8D 4C $1ｦ■.▌ｧ■
4E88 08 89 6A 8D 08 86 A6 8D 49 .▌j■.■ｦ■
4E90 08 82 A1 8D 08 79 24 1B 78 .▂｡■.y$.
4E98 6D 8D 08 76 27 0D 17 02 C5 m■.v´...
4EA0 50 A6 8D 08 6A A7 8D 08 31 Pｦ■.jｧ■.
4EA8 6B 20 08 A6 8D 08 61 A7 D6 k .ｦ■.aｧ
4EB0 8D 08 61 EC 8D 08 5C 17 EA ■.a●■.¥.
4EB8 03 B7 39 6A 8D 08 55 A6 ED .ｷ9j■.Uｦ
4EC0 8D 08 51 A1 8D 08 48 24 88 ■.Q｡■.H$
4EC8 1B 6D 8D 08 45 27 0D 17 AD .m■.E´..
4ED0 02 1F A6 8D 08 39 A7 8D C9 ..ｦ■.9ｧ■
4ED8 08 3A 20 08 A6 8D 08 30 D5 .: .ｦ■.0
4EE0 A7 8D 08 30 EC 8D 08 2B 18 ｧ■.0●■.+
4EE8 17 03 86 39 6C 8D 08 24 FE ..■9l■.$
4EF0 A6 8D 08 20 A1 8D 08 18 A9 ｦ■. ｡■..
4EF8 23 1B 6D 8D 08 14 27 0D 88 #.m■..´.
-----------------------------------------
SUM  25 54 8F 75 D1 03 6A 0F CA
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4F00 17 01 75 A6 8D 08 09 A7 78 ..uｦ■..ｧ
4F08 8D 08 09 20 08 A6 8D 07 00 ■.. .ｦ■.
4F10 FE A7 8D 07 FF EC 8D 07 B8 ｧ■..●■.
4F18 FA 17 03 55 39 A6 8D 07 DC ..U9ｦ■.
4F20 F2 A0 8D 07 E8 81 07 23 B9 年.■.♠▁.#
4F28 01 4A C6 03 3D C3 00 05 19 .Jﾆ.=ﾃ..
4F30 1F 98 E6 8D 07 DE 17 03 29 .┌◥.ﾞ..
4F38 38 39 34 02 E6 8D 07 D4 F5 894.◥.ﾔ
4F40 E0 8D 07 CB A6 8D 07 B6 2F ═■.ﾋｦ■.ｶ
4F48 81 08 23 01 58 58 58 58 0D ▁.#.XXXX
4F50 4F ED 8D 07 A4 35 02 4D F8 O○■.､5.M
4F58 26 07 17 FE 36 1F 89 20 40 &.. 6.▌
4F60 1B A6 8D 07 99 81 08 22 99 .ｦ■.┐▁."
4F68 08 E6 8D 07 A6 C0 20 20 28 .◥.ｦﾀ
4F70 0B E6 8D 07 9E C0 38 C1 DC .◥.╰ﾀ8ﾁ
4F78 08 25 01 5A 4F E3 8D 07 4E .%.ZO╡■.
-----------------------------------------
SUM  F2 A2 F1 FB E3 0C AC 40 5B
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4F80 78 AE 8D 07 70 30 8B 39 1E xｮ■.p0▌9
4F88 17 FE 08 5D 27 07 A6 8D DB . .]´.ｦ■
4F90 07 81 4A 20 04 A6 8D 07 30 .▁J .ｦ■.
4F98 7A E6 8D 07 77 17 02 D1 55 z◥.w..ﾑ
4FA0 4F 8D 97 A6 84 17 02 17 CD O■ ■ｦ▄...
4FA8 86 07 17 02 E9 A6 8D 07 C9 ■...♥ｦ■.
4FB0 62 E6 8D 07 5F 17 02 B9 0D b◥._..ｹ
4FB8 39 17 FD D7 34 02 A6 8D 8D 9. ﾗ4.ｦ■
4FC0 07 3C 81 08 22 05 35 02 2A .<▁.".5.
4FC8 8B 20 39 35 02 8B 38 81 5F ▌ 95.▌8▁
4FD0 40 24 01 39 4C 39 17 FF 39 @$.9L9..
4FD8 61 A6 84 17 F9 02 17 02 B6 aｦ▄. ...
4FE0 AC 39 17 FF 55 34 10 E6 7A ｬ9..U4.◥
4FE8 8D 07 29 E0 8D 07 20 A6 F7 ■.)═■. ｦ
4FF0 8D 07 0B 81 08 23 01 58 A4 ■..▁.#.X
4FF8 58 58 58 4F 10 AE 8D 06 A8 XXXO.ｮ■.
-----------------------------------------
SUM  D1 69 86 4D 75 A1 50 70 E3
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5000 F4 31 AB 34 20 A6 8D 06 5D 日1ｫ4 ｦ■.
5008 F5 17 FD 1E 34 02 A6 8D 90 ■. .4.ｦ■
5010 06 EC 80 08 A7 8D 06 E1 95 .●▁.ｧ■.╞
5018 48 48 A0 8D 06 DB 8B 1D 46 HH.■.ﾛ▌.
5020 E6 8D 06 F0 17 02 4A 86 52 ◥.X..J■
5028 05 17 02 6A 35 02 17 01 D7 ...j5...
5030 8E 86 07 17 02 60 35 20 E9 ■■...`5
5038 10 AF 8D 06 DD 35 06 A3 0D .ｯ■.ﾝ5.｣
5040 8D 06 D7 ED 8D 06 D3 AE 6B ■.ﾗ○■.ﾓｮ
5048 8D 06 AA 30 8B 17 FC EE F9 ■.ｪ0▌. ◢
5050 34 02 EC 8D 06 C4 58 58 29 4.●■.ﾄXX
5058 E0 8D 06 BF CB 05 1F 98 B9 ═■.ｿﾋ..┌
5060 C6 14 17 02 0C 86 05 17 A1 ﾆ....■..
5068 02 2C 35 02 17 01 50 86 53 .,5...P■
5070 07 17 02 22 17 01 02 39 95 ..."...9
5078 E6 8D 06 82 4F ED 8D 06 CA ◥.▂O○■.
-----------------------------------------
SUM  A3 D4 2B 6F 9E 04 8A 43 80
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5080 78 EC 8D 06 70 E3 8D 06 DD x●■.p╡■.
5088 70 ED 8D 06 68 ED 8D 06 D8 p○■.h○■.
5090 73 86 04 A7 8D 06 7F C6 7C s■.ｧ■..ﾆ
5098 0F 34 04 17 00 BD 34 26 75 .4...ｽ4&
50A0 17 02 91 31 8D 06 D0 17 55 ..┬1■.ﾐ.
50A8 02 A1 35 06 5A ED 8D 06 B8 .｡5.Z○■.
50B0 48 35 06 5A 1F 02 EC 8D 77 H5.Z..●■
50B8 06 3F 30 8D 06 B9 17 02 DA .?0■.ｹ..
50C0 CE 6C 8D 06 51 35 04 5A B1 ﾎl■.Q5.Z
50C8 26 CF CC 00 12 17 01 A1 8C &ﾏﾌ....｡
50D0 E6 8D 06 2A 4F ED 8D 06 72 ◥.*O○■.
50D8 20 EC 8D 06 18 E3 8D 06 2D  ●■..╡■.
50E0 16 A3 8D 06 14 1F 01 17 97 .｣■.....
50E8 F6 A7 17 02 2D 17 F8 75 67 分ｧ..-. u
50F0 39 E6 8D 06 09 4F ED 8D 84 9◥..O○■
50F8 05 FF EC 8D 05 F7 A3 8D A9 ..●■.秒｣■
-----------------------------------------
SUM  15 8D 27 B9 8A D9 D5 51 0B
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5100 05 F7 ED 8D 05 EF ED 8D E4 .秒O■.\O■
5108 05 FA 86 11 A7 8D 06 06 D6 .■.ｧ■..
5110 C6 0F 34 04 8D 45 34 26 39 ﾆ.4.■E4&
5118 17 02 19 31 8D 06 58 17 65 ..1■.X.
5120 02 29 35 06 5C ED 8D 05 41 .)5.¥O■.
5128 D0 35 06 5C 1F 02 EC 8D 01 ﾐ5.¥..●■
5130 05 C7 30 8D 06 41 17 02 E9 .ﾇ0■.A..
5138 56 6A 8D 05 D9 35 04 5A BE Vj■.ﾙ5.Z
5140 26 D0 CC 00 03 17 01 29 06 &ﾐﾌ....)
5148 EC 8D 05 A9 1F 01 17 F6 54 ●■.ｩ...个
5150 40 CC 00 14 17 01 1A 17 69 @ﾌ.....
5158 F8 0B 39 4F E6 8D 05 B6 B9 .9O◥■.ｶ
5160 34 06 A6 8D 05 98 81 08 93 4.ｦ■.┏_
5168 23 04 86 48 20 02 86 27 C4 #.■H .■
5170 E6 8D 05 A2 1F 02 35 06 76 ◥■.｢.5.
5178 39 E6 8D 05 81 AE 8D 05 72 9◥■._ｮ■.
-----------------------------------------
SUM  D4 42 80 4F 04 1C 13 E4 FC
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5180 74 34 04 34 10 17 FB B6 B8 t4.4.. ｶ
5188 36 02 35 10 1F 10 C3 00 6F 6.5...ﾃ.
5190 01 1F 01 35 04 5A 26 E9 C3 ...5.Z&♥
5198 17 FB E1 34 02 A6 8D 05 61 .ﾄ4.ｦ■.
51A0 5D 81 08 23 04 86 35 20 E8 ]_.#.■5
51A8 02 86 1D C6 14 17 00 C1 57 .■.ﾆ...ﾁ
51B0 86 05 17 00 E1 35 02 8D 47 ■...ﾄ5.■
51B8 06 86 07 17 00 D8 39 17 D2 .■..ﾘ9.
51C0 02 68 ED 8D 05 AC 86 FF 1A .hO■.ｬ■.
51C8 A7 8D 05 A8 31 8D 05 A2 46 ｧ■.ｨ1■.｢
51D0 17 00 F3 39 86 07 17 00 E7 ..月9■...
51D8 BD 8D 72 0D 0A 20 41 62 96 ｽ■r.. Ab
51E0 6F 72 74 65 64 0D 0A 0D 42 orted...
51E8 0A FF 30 8D EE 2E AF 6A FB ..0■/.ｯj
51F0 3B B6 FD 05 2B FB 86 80 1F ;ｶ.+ ■_
51F8 B7 FD 05 B6 FD 05 2A FB 96 ｷ.ｶ.*
-----------------------------------------
SUM  95 88 5B D5 6E 6C 2D 1E 72
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5200 39 4F B7 FD 05 39 B6 FC 2C 9Oｷ .9ｶ
5208 80 8A 80 B7 FC 80 8D F1 3B _▙_ｷ _■円
5210 39 8D DE 8E FC 82 86 0A 40 9■ﾞ■ ▂■.
5218 A7 84 8D E5 39 34 02 8D 99 ｧ▃■◢94.■
5220 D0 8E FC 82 86 01 A7 80 8A ﾐ■ ▂■.ｧ_
5228 86 00 A7 80 A6 8D 04 BE A2 ■.ｧ_ｦ■.ｾ
5230 A7 80 86 19 A7 80 6F 80 DC ｧ_■.ｧ_o_
5238 86 19 A7 80 6F 80 86 FF 3A ■.ｧ_o_■.
5240 A7 80 35 02 A7 84 8D B9 CF ｧ_5.ｧ▃■ｹ
5248 86 07 8D 4A 39 35 10 8D 6F ■.■J95.■
5250 A0 10 8E FC 81 6F A0 86 50 ..■ _o ■
5258 03 A7 A1 5F A6 80 81 FF 50 .ｧ｡_ｦ__
5260 27 05 A7 A0 5C 20 F5 10 F4 '.ｧ ¥ 个.
5268 8E FC 83 E7 A4 8D 92 1F D6 ■ ▄◤.■..
5270 15 ED 8D 00 03 8D D6 12 07 .O■..■ﾖ.
5278 00 08 FF 39 34 02 86 1C 18 ...94.■.
-----------------------------------------
SUM  B6 45 19 29 B6 E1 0C 69 49
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5280 8D 0B 35 02 4A 26 F5 39 6D ■.5.J&个9
5288 86 1D 8D 01 39 A7 8D 00 9E ■.■.9ｧ■.
5290 02 8D BA 3D FF 39 A7 8D F2 .■ｺ=.9ｧ■
5298 00 03 8D B1 11 07 FF 39 91 ..■ｱ...9
52A0 17 FF 4E 8E FC 82 86 29 1F ..N■ ▂■)
52A8 A7 80 6F 84 17 FF 52 17 99 ｧ_o▃..R.
52B0 FF 3F 8E FC 83 E6 80 A6 57 .?■ ▄◥_ｦ
52B8 84 34 02 17 FF 48 35 02 4F ▃4...H5.
52C0 39 86 07 8D C8 39 34 20 A8 9■.■ﾈ94
52C8 17 FF 26 8E FC 81 6F 80 36 ..&■ _o_
52D0 86 03 A7 81 5F A6 A0 81 D7 ■.ｧ__ｦ _ﾗ
52D8 FF 27 05 A7 80 5C 20 F5 C3 .'.ｧ_¥ 个
52E0 8E FC 83 E7 84 17 FF 19 A7 ■ ▄◤▃...
52E8 35 20 39 86 0C 8D 9E 39 84 5 9■.■ 9
52F0 86 0E A7 8D 06 1D 10 AF AA ■.ｧ■....ｯ
52F8 8D 06 1A 5F A6 A0 81 FF D2 ■.._ｦ _.
-----------------------------------------
SUM  01 89 AC B2 07 D9 46 FD 0B
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5300 27 03 5C 20 F7 4F ED 8D 66 '.¥ 秒O■
5308 06 0D 30 8D 06 05 AD 9F 27 ..0■..ｭ'
5310 FB FA 39 31 8D 05 AD 8D 2B   91■.ｭ■
5318 D7 39 31 8D 05 A6 8D A6 AC ﾗ91■.ｦ■ｦ
5320 39 17 FE ED 17 FE CA 8E A8 9. O. ﾊ■
5328 FC 83 EC 84 34 06 17 FE 3E  ▄●▃4..
5330 D5 35 06 39 34 06 17 FE 98 ﾕ5.94..
5338 B8 8E FC 82 86 08 A7 80 79 ｸ■ ▂■.ｧ_
5340 35 06 ED 81 10 AF 84 17 03 5.O_.ｯ▃.
5348 FE B7 39 17 FE A3 F6 FC 98  ｷ9. ｣个
5350 83 8E FC 84 A6 80 A7 A0 FE ▄■ ▃ｦ_ｧ .
5358 5A 26 F9 F6 FC 81 17 FE 01 Z& 个 _.
5360 A5 C1 80 25 25 17 FE 89 CE ･ﾁ_%%. ▎
5368 86 64 B7 FC 82 17 FE 91 C5 ■dｷ ▂. ┬
5370 17 FE 7E F6 FC 83 8E FC 92 . ~个 ▄■
5378 84 A6 80 A7 A0 5A 26 F9 6A ▃ｦ_ｧ Z&
-----------------------------------------
SUM  97 DA 32 67 87 6F 5B 29 84
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5380 F6 FC 81 17 FE 80 C1 80 49 个 _. _ﾁ_
5388 24 DB 86 FF A7 A4 39 34 3C $ﾛ■.ｧ､94
5390 06 17 FE 5D 86 09 B7 FC BA .. ]■.ｷ
5398 82 35 06 FD FC 83 10 BF 08 ▂5.  ▄.ｿ
53A0 FC 85 5F 10 8E FC 89 A6 A9  ▅_.■ ▎ｦ
53A8 80 81 FF 27 15 A7 A0 5C DF __.'.ｧ ¥
53B0 C1 77 25 F3 B6 FC 81 8A 0D ﾁw%月ｶ _▙
53B8 80 B7 FC 81 A7 8D 03 39 24 _ｷ _ｧ■.9
53C0 20 0C B6 FC 81 84 7F B7 19  .ｶ _▃.ｷ
53C8 FC 81 6F 8D 03 2B F7 FC 9A  _o■.+秒
53D0 88 17 FE 2D 6D 8D 03 21 E8 ▎. -m■.!
53D8 27 3C 17 FE 14 86 64 B7 2D '<. .■dｷ
53E0 FC 82 10 8E FC 84 5F A6 A1  ▂.■ ▃_ｦ
53E8 80 81 FF 27 15 A7 A0 5C DF __.'.ｧ ¥
53F0 C1 7C 25 F3 B6 FC 81 8A 12 ﾁ!%月ｶ _▙
53F8 80 B7 FC 81 A7 8D 02 F9 E3 _ｷ _ｧ■.
-----------------------------------------
SUM  E7 6D F4 F8 9A 52 CD 44 3D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5400 20 0C B6 FC 81 84 7F B7 19  .ｶ _▃.ｷ
5408 FC 81 6F 8D 02 EB F7 FC 59  _o■.■秒
5410 83 17 FD ED 20 BE 39 34 CF ▄.  O ｾ94
5418 20 34 06 8D 0D ED A1 35 B7  4.■.O｡5
5420 06 1F 98 8D 05 ED A4 35 15 ..┤■.O､5
5428 20 39 34 02 44 44 44 44 9F  94.DDDD
5430 8B 30 81 3A 25 02 8B 07 2F ▎0_:%.▎.
5438 35 04 C4 0F CB 30 C1 3A 02 5.ﾄ.ﾋ0ﾁ:
5440 25 02 CB 07 39 34 10 E6 5C %.ﾋ.94.◥
5448 80 C1 30 25 1B C1 39 23 CE _ﾁ0%.ﾁ9#
5450 10 C1 41 25 13 C1 46 23 74 .ﾁA%.ﾁF#
5458 08 C1 61 25 0B C1 66 22 A3 .ﾁa%.ﾁf"
5460 07 4A 26 E3 1C FE 20 02 96 .J&┤. .
5468 1A 01 35 10 39 34 10 EC C9 ..5.94.●
5470 81 8D 0D 34 02 EC 84 8D 4E _■.4.●▃■
5478 07 1F 89 35 02 35 10 39 64 ..▎5.5.9
-----------------------------------------
SUM  0B A0 C7 AD B4 47 3D D8 2F
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5480 34 04 81 39 23 02 80 07 9E 4._9#._.
5488 84 0F 48 48 48 48 35 04 EC ▃.HHHH5.
5490 C1 39 23 02 C0 07 C4 0F B9 ﾁ9#.ﾀ.ﾄ.
5498 34 04 AA E4 35 04 39 6F A7 4.ｪ◢5.9o
54A0 8D 02 58 17 FD FA 81 00 76 ■.X.  _.
54A8 27 F9 C1 03 26 04 86 FF 93 ' ﾁ.&.■.
54B0 20 51 C1 0D 27 46 C1 20 8D  Qﾁ.'Fﾁ
54B8 25 21 C1 7F 27 E5 30 8D 4F %!ﾁ.'◣0■
54C0 02 71 A6 8D 02 35 E7 86 4A .qｦ■.5◤■
54C8 1F 98 17 FD C0 6C 8D 02 86 .┤. ﾀl■.
54D0 2A A6 8D 02 26 81 40 27 6D *ｦ■.&_@'
54D8 23 20 C8 C1 1D 27 06 C1 D7 # ﾈﾁ.'.ﾁ
54E0 08 27 02 20 BE A6 8D 02 44 .'. ｾｦ■.
54E8 12 27 B8 6A 8D 02 0C 17 0D .'ｸj■...
54F0 FD 96 86 20 17 FD 96 17 FA  ▎■ . ▎.
54F8 FD 8E 20 A7 17 FE 1B A6 28  ■ ｧ. .ｦ
-----------------------------------------
SUM  28 FE A3 AB 4F 6A AE 7B 56
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5500 8D 01 F8 A7 8D 02 2B 39 20 ■. ｧ■.+9
5508 EC 42 26 07 37 06 37 06 D5 ●B&.7.7.
5510 1A 01 39 CC 00 00 34 06 5A ..97..4.
5518 EC C4 10 A3 42 25 0D 37 0E ●ﾄ.｣B%.7
5520 06 A3 C4 36 06 35 06 C3 A7 .｣ﾄ6.5.ﾃ
5528 00 01 20 EA 37 06 37 20 9F .. ◆7.7
5530 36 06 35 06 36 06 1C FE CD 6.5.6..
5538 39 34 10 34 06 34 02 A6 93 94.4.4.ｦ
5540 80 A1 62 27 0A 35 02 4A 35 _｡b'.5.J
5548 26 F3 35 10 35 10 39 A6 82 &円5.5.9ｦ
5550 61 A0 E4 4C 35 04 35 10 AF a.◢L5.5.
5558 35 10 39 8D 44 25 34 4A F2 5.9■D%4J
5560 26 03 4F 20 39 34 02 4F 56 &.O 94.O
5568 34 06 8D 35 25 28 4F 34 CC 4.■5%(O4
5570 06 EC 62 58 49 58 49 58 EE .●bXIXIX
5578 49 E3 62 E3 62 25 0E E3 E9 I┤b┤b%.┤
----------------------------------------
SUM  D9 02 E4 17 40 E9 4A 0B 54

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5580 E4 25 0A 35 20 35 20 6A 27 ◢%.5 5 j
5588 E4 26 DD 20 0B 35 06 35 82 ◢&ﾝ .5.5
5590 06 35 04 1A 01 39 35 06 CE .5...95.
5598 1F 02 35 04 1F 20 1C FE B3 ..5.. .
55A0 39 E6 80 C1 30 25 09 C1 7F 9◥_ﾁ0%.ﾁ
55A8 39 22 05 C4 0F 1C FE 39 86 9".ﾄ.. 9
55B0 1A 01 39 34 10 34 06 CC 9E ..94.4.7
55B8 00 0A 36 06 35 06 36 06 BD ..6.5.6.
55C0 17 FF 45 37 06 34 06 37 09 ..E7.4.7
55C8 06 CA 30 E7 80 35 06 10 B2 .ﾊ0◤_5..
55D0 83 00 00 26 E0 1F 10 A3 5B ▂..&=..｣
55D8 E4 35 20 39 EC C4 10 A3 D5 ◢5 9●ﾄ.｣
55E0 42 24 08 37 10 37 06 36 28 B$.7.7.6
55E8 10 36 06 EC 42 10 83 00 0D .6.●B.▂.
55F0 00 27 12 CC 00 00 E3 C4 AC .'.7..┤ﾄ
55F8 34 06 EC 42 83 00 01 ED D9 4.●B▂..○
----------------------------------------
SUM  83 1A B5 E0 F6 D1 53 E3 2F

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5600 42 35 06 26 F1 37 10 37 12 B5.&月7.7
5608 10 39 8D 5A 25 25 34 02 B0 .9■Z%%4.
5610 CC 00 00 36 06 35 02 34 73 7..6.5.4
5618 02 E6 80 C4 0F EA 41 E7 4D .◥_ﾄ.◆A◤
5620 41 68 41 69 C4 35 02 4A 98 AhAiﾄ5.J
5628 26 ED 66 C4 66 41 37 06 21 &○fﾄfA7.
5630 1C FE 39 39 34 10 4D 26 43 . 994.M&
5638 08 1F 98 34 06 86 08 20 A7 ..┌4.■.
5640 04 34 06 86 10 36 02 35 41 .4.■.6.5
5648 06 58 49 25 06 34 06 86 92 .XI%.4.■
5650 30 20 04 34 06 86 31 A7 EC 0 .4.■1ｧ
5658 80 6A C4 26 EA 35 06 1F 18 _jﾄ&◆5..
5660 10 A3 E4 35 10 39 34 12 5B .｣◢5.94.
5668 E6 80 C1 30 27 06 C1 31 76 ◥_ﾁ0'.ﾁ1
5670 27 02 20 07 4A 26 F1 1C CD '. .J&月.
5678 FE 20 02 1A 01 35 12 39 BB  ...5.9
----------------------------------------
SUM  80 21 69 9F 17 46 4C 03 55

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5680 34 34 A6 80 A1 A0 26 07 FC 44ｦ_｡.&.
5688 5A 26 F7 1A 01 20 02 1C D0 Z&秒.. ..
5690 FE 35 34 39 34 10 34 02 1A 5494.4.
5698 A6 80 81 61 25 08 81 7A 30 ｦ__a%._z
56A0 22 04 80 20 A7 1F 35 02 C3 ".ｧ_ ｧ.5.
56A8 4A 26 EB 35 10 39 EA 54 17 J&♣5.9◆T
56B0 EA CC 03 7D 05 0F 06 54 A4 ◆7.}...T
56B8 08 F2 0C 78 0F 9F 10 FF 3B .年.x.┘..
56C0 00 00 00 00 00 00 00 00 00 ........
56C8 00 00 00 00 00 00 44 64 A8 ......Dd
56D0 57 77 54 74 46 66 43 63 E8 WwTtFfCc
56D8 53 73 52 72 47 67 42 62 DC SsRrGgBb
56E0 00 00 00 00 00 00 00 00 00 ........
56E8 00 00 00 00 00 00 28 00 28 ......(.
56F0 08 40 00 68 FF 56 80 00 85 .@.h.V_.
56F8 80 00 08 02 00 01 08 01 94 _.......
----------------------------------------
SUM  C2 21 7A CE 52 01 8B 72 7B

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5700 08 72 00 00 68 FF 00 00 67 .r..h...
5708 00 00 5A 1F 00 00 00 00 79 ..Z.....
5710 00 00 00 00 00 00 00 00 00 ........
5718 00 00 00 00 00 00 00 00 00 ........
5720 00 00 00 00 00 00 00 00 00 ........
5728 00 00 00 00 00 00 00 00 00 ........
5730 00 00 02 2F 3D 46 46 00 FA .../=FF.
5738 00 00 00 00 00 00 00 00 00 ........
5740 00 00 00 00 00 00 00 00 00 ........
5748 00 00 00 00 00 00 00 00 00 ........
5750 00 00 00 00 00 00 00 00 00 ........
5758 00 00 00 00 00 00 00 00 00 ........
5760 00 00 00 00 00 00 00 00 00 ........
5768 00 00 00 00 00 00 00 00 00 ........
5770 00 00 30 30 20 FF FF 35 B3 ..00 ..5
5778 37 37 38 20 33 37 20 33 83 778 37 3
----------------------------------------
SUM  2D 20 C6 B1 E5 87 7B 52 FD

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5780 42 31 20 30 20 33 30 20 56 B1 0 30
5788 32 30 20 33 33 20 33 33 6E 20 33 33
5790 20 33 33 20 FF 2D 2D 2D 3C  33 .---
5798 2D 2D 2D 2D 2D 2D 2D FF 3A -------.
57A0 00 07 00 07 00 07 00 07 1C ........
57A8 00 07 00 07 00 07 00 07 1C ........
57B0 00 07 00 07 00 07 00 07 1C ........
57B8 00 07 00 07 00 07 00 07 1C ........
57C0 00 07 00 07 00 07 00 07 1C ........
57C8 00 07 00 07 00 07 00 07 1C ........
57D0 00 07 00 07 00 07 00 07 1C ........
57D8 00 07 00 07 00 07 00 07 1C ........
57E0 00 07 00 07 00 07 00 07 1C ........
57E8 00 07 00 07 00 07 00 07 1C ........
57F0 00 07 00 07 00 07 00 07 1C ........
57F8 00 07 00 07 00 07 00 07 1C ........
----------------------------------------
SUM  B4 08 B4 08 B4 FB CD EF E3

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5800 00 07 00 07 00 07 00 07 1C ........
5808 00 07 00 07 00 07 00 07 1C ........
5810 00 07 00 07 00 07 00 FF 14 ........
5818 00 00 00 00 00 00 00 00 00 ........
5820 00 00 00 00 00 00 00 00 00 ........
5828 00 00 00 00 00 00 00 00 00 ........
5830 00 00 00 00 00 00 00 20 20 .......
5838 44 55 4D 50 49 4E 47 20 34 DUMPING
5840 5B 34 30 30 30 2D 36 38 BA [4000-68
5848 46 46 5D 0D 0A FF 0D 0A 16 FF].....
5850 41 44 44 52 20 2B 30 20 B6 ADDR +0
5858 2B 31 20 2B 32 20 2B 33 57 +1 +2 +3
5860 20 2B 34 20 2B 35 20 2B 4A  +4 +5 +
5868 36 20 2B 37 20 3D 53 20 88 6 +7 =S
5870 30 31 32 33 34 35 36 37 9C 01234567
5878 FF 0D 0A 41 44 44 52 20 51 ...ADDR
----------------------------------------
SUM  D6 E2 D9 EA 98 C5 E0 84 3C

ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5880 2B 30 20 2B 31 20 2B 32 54 +0 +1 +2
5888 20 2B 33 20 2B 34 20 2B 48  +3 +4 +
5890 35 20 2B 36 20 2B 37 20 58 5 +6 +7
5898 2B 38 20 2B 39 20 2B 41 73 +8 +9 +A
58A0 20 2B 42 20 2B 43 20 2B 66  +B +C +
58A8 44 20 2B 45 20 2B 46 20 85 D +E +F
58B0 3D 53 20 30 31 32 33 34 AA =S 01234
58B8 35 36 37 20 38 39 41 42 B6 567 89AB
58C0 43 44 45 46 0D 0A FF 53 7B CDEF...S
58C8 55 4D 20 20 FF 20 47 6F B7 UM  . Go
58D0 20 20 41 64 64 72 20 FF DA   Addr .
58D8 20 42 72 6B 20 41 64 64 68  Brk Add
58E0 72 20 FF 2B 2D 48 68 41 DA r .+-HhA
58E8 61 41 61 2A 42 62 2F 43 43 aAa*Bb/C
58F0 63 2B 44 64 2D 45 65 3D 4A c+Dd-Ee=
58F8 46 66 2C 2B 2D 2A 2F 43 CC Ff,+-*/C
----------------------------------------
SUM  D5 6C 4A 7A C2 6E 7C A8 59
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5900 43 41 20 42 20 44 50 44 DE CA B DPD
5908 20 58 20 59 20 55 20 50 D6  X Y U P
5910 43 53 50 0E 00 58 C4 00 C2 CSP..Xﾄ.
5918 02 00 00 00 00 00 00 00 1D ........
5920 00 00 00 00 00 00 00 00 00 ........
5928 00 00 00 00 00 00 00 00 00 ........
5930 00 00 00 00 00 00 00 00 00 ........
5938 00 00 00 00 00 00 00 00 00 ........
5940 00 00 00 00 00 00 00 00 00 ........
5948 00 00 00 00 00 00 00 00 00 ........
5950 00 00 00 00 00 00 00 00 00 ........
5958 00 00 00 00 00 00 00 00 00 ........
5960 00 00 00 00 00 00 00 00 00 ........
5968 00 00 00 00 00 00 00 00 00 ........
5970 00 00 00 00 00 00 00 00 00 ........
5978 00 00 00 00 00 00 00 00 00 ........
----------------------------------------
SUM  AB EC 90 A9 40 F1 37 94 CC
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5980 00 00 00 00 00 00 00 00 00 ........
5988 00 00 00 00 00 00 00 00 00 ........
5990 00 00 00 00 00 00 00 00 00 ........
5998 00 00 00 A0 00 00 00 40 E0 .......@
59A0 00 80 72 80 00 40 0B 3E FB ._r_.@.>
59A8 CF 00 00 FF FF 00 00 00 CD ﾏ.......
59B0 00 00 00 00 00 00 00 00 00 ........
59B8 00 00 00 00 00 00 00 00 00 ........
59C0 00 00 00 00 00 00 00 00 00 ........
59C8 00 00 00 00 00 00 00 00 00 ........
59D0 00 00 00 00 00 00 00 00 00 ........
59D8 00 00 00 00 00 00 00 00 00 ........
59E0 00 00 00 00 00 00 00 00 00 ........
59E8 00 00 00 00 00 00 00 00 00 ........
59F0 00 00 00 00 00 00 00 00 00 ........
59F8 00 00 00 00 1C F5 1C F5 22 ....→時→時
----------------------------------------
SUM  EB 80 72 9F 06 2A 43 9D 8C
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5A00 1C F5 B6 F1 9A 59 13 A0 3E .時ｶ円└Y..
5A08 00 59 13 58 C7 59 5A 54 CE .Y.XｷYZT
5A10 10 54 25 57 77 48 48 47 76 .T%WwHHG
5A18 A6 44 74 41 11 40 2C A0 DD ｦDtA.@,.
5A20 00 00 00 40 00 80 72 80 B2 ...@._r_
5A28 00 40 0B 8D 3F 17 E8 99 AF .@.■?.♠
5A30 8D 54 17 00 A1 4D 27 17 24 ■T...｡M'.
5A38 86 01 A7 8D FC C4 17 EA 7C ■.ｧ■ ﾄ.◆
5A40 26 EC 8D FC D2 ED 8D FC E3 &●■ ﾒ○■
5A48 AC 17 EA 1B 17 00 A4 17 9A ｬ.◆.....
5A50 00 F9 6D 8D FC C3 26 05 DD . m■ ﾃ&.
5A58 17 F8 BF 20 06 17 01 49 55 . ｿ ...I
5A60 17 EA 04 6F 8D FC 9B 6F 07 .◆.o■ ┘o
5A68 8D FC AE 39 17 F7 DE 0D 69 ■ ｮ9.秒ﾞ.
5A70 0A 3E 42 4C 4F 43 4B 20 D3 .>BLOCK
5A78 54 52 41 4E 53 46 45 52 65 TRANSFER
----------------------------------------
SUM  46 9B 24 D7 58 09 04 BB FC
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5A80 0D 0A 0D 0A FF 39 17 F7 74 .....9.秒
5A88 C4 20 44 65 73 74 20 20 B4 ﾄ Dest
5A90 41 64 64 72 20 FF EC 8D 13 Addr .●■
5A98 FC 7D 31 8D FC D4 17 F9 17 )1■ ﾄ.
5AA0 76 17 F8 22 17 F7 A6 20 7B v.".秒ｦ
5AA8 3F FF 17 F9 F2 30 8D FC F9 ?.. 年0■
5AB0 81 A6 80 81 FF 26 04 4F A0 _ｦ__.&.O
5AB8 17 F2 B0 4D 27 17 81 04 C9 .年ｰM'._.
5AC0 26 0E 17 F9 80 25 09 17 09 &.. _%..
5AC8 F9 A3 ED 8D FC 49 20 05 80 ｣○■ I .
5AD0 17 ED 82 20 B1 39 17 F7 9E .○_ ｱ9.秒
5AD8 74 20 56 65 72 69 66 79 09 t Verify
5AE0 20 28 59 2F 45 4C 53 45 F9  (Y/ELSE
5AE8 29 3F FF 17 EF EB A7 8D 8C )?..◣ｧ■
5AF0 FC 27 39 17 F7 57 20 4F 30 '9.秒W O
5AF8 4B 20 3F FF 17 E6 C0 17 7D K ?..◥ﾀ.
----------------------------------------
SUM  95 25 D1 BE 9E 68 72 D0 91
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5B00 ED 43 86 04 17 F7 8F 17 6E ○C■..秒┼.
5B08 F7 43 20 52 45 54 3A 6F EE 秒C RET:o
5B10 6B 28 63 6F 6E 74 29 20 90 k(cont)
5B18 43 54 52 4C 2D 43 3A 61 40 CTRL-C:a
5B20 62 6F 72 74 FF EC 8D FB 2A bort.●■
5B28 C6 17 F7 45 86 07 17 F7 B4 ﾆ.秒E■..秒
5B30 65 17 F7 6C 4D 27 FA C1 0E e.秒lM' ﾁ
5B38 03 26 05 86 01 17 F2 2B E9 .&.■..年+
5B40 C1 0D 26 ED 17 E6 9A 17 8F ﾁ.&○.◥└.
5B48 F7 D0 39 EC 8D FB A4 A3 BB 秒ﾐ9●■ ､｣ｻ
5B50 8D FB 9E C3 00 01 ED 8D 64 ■ ﾃ..○■
5B58 FB C2 34 06 EC 8D FB B7 22 ﾂ4.●■ ｷ
5B60 10 A3 8D FB 8C 25 29 10 25 .｣■ ■%).
5B68 A3 8D FB 87 22 22 EC 8D 6F ｣■ ■""●■
5B70 FB 81 C3 00 01 1F 01 EC 4C _ﾃ....●
5B78 8D FB 9C E3 E4 1F 02 35 41 ■ ◢◢..5
----------------------------------------
SUM  9D 0B D8 C3 ED 27 FA A1 F2
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5B80 06 34 06 A6 82 A7 A2 35 E6 .4.ｦ_ｧ｢5
5B88 06 83 00 01 26 F3 20 18 DB .■..&月 .
5B90 AE 8D FB 5D 10 AE 8D FB D9 ｮ■ ].ｮ■
5B98 7E 35 06 34 06 A6 80 A7 C0 ~5.4.ｦ_ｧ
5BA0 A0 35 06 83 00 01 26 F3 78 .5.■..&月
5BA8 39 EC 8D FB 6A ED 8D FB 8C 9●■ j○■
5BB0 44 E3 8D FB 67 83 00 01 9A D┤■ g■..
5BB8 ED 8D FB 48 39 8D 31 17 CB ○■ H9■1.
5BC0 E7 07 8D 49 17 FF 0F 4D 36 ◤.■I...M
5BC8 27 0C 86 01 A7 8D FB 32 1B '.■.ｧ■ 2
5BD0 17 E8 94 17 FF 1D 17 01 DE .♠¯.....
5BD8 02 6D 8D FB 3C 26 05 17 75 .m■ <&..
5BE0 F7 38 20 03 17 E8 80 6F 40 秒8 ..♠_o
5BE8 8D FB 17 6F 8D FB 2A 39 F9 ■ .o■ *9
5BF0 17 F6 5A 0D 0A 3E 46 49 4B .分Z..>FI
5BF8 4C 4C 20 4D 45 4D 4F 52 38 LL MEMOR
----------------------------------------
SUM  50 E7 07 21 B4 29 18 CF 23
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5C00 59 20 42 4C 4F 43 4B 0D F1 Y BLOCK.
5C08 0A 0D 0A FF 39 17 F6 3D A3 ....9.分=
5C10 20 46 69 6C 6C 20 56 61 7E  Fill Va
5C18 6C 75 65 20 3F FF 17 E5 A0 lue ?..◣
5C20 9E 86 0C 17 F6 70 17 F6 BA ◥■..分p.分
5C28 24 56 41 4C 55 45 3A FF DA $VALUE:.
5C30 86 04 17 F6 61 17 F6 15 1A ■..分a.分.
5C38 20 48 45 58 3A 32 28 30 C9  HEX:2(0
5C40 30 2D 46 46 29 20 44 45 BB 0-FF) DE
5C48 43 49 4D 41 4C 3A 27 23 EA CIMAL:'#
5C50 27 2B 28 30 2D 32 35 35 73 '+(0-255
5C58 29 20 43 48 41 52 3A 27 C8 ) CHAR:'
5C60 22 27 2B 41 53 43 49 49 DD "'+ASCII
5C68 20 43 48 41 52 FF 86 07 CA  CHAR.■.
5C70 17 F6 23 EC 8D FA 78 17 32 .分#●■ x.
5C78 F5 F7 17 F8 22 30 8D FA D4 時秒. "0■
----------------------------------------
SUM  68 28 6E ED 50 C1 CB EF B6
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5C80 B1 A6 80 81 FF 26 05 86 08 ｱｦ__.&.■
5C88 01 17 F0 DF 81 02 25 42 D1 ..×ﾟ_.%B
5C90 34 02 A6 80 81 22 26 0A 2F 4.ｦ__"&.
5C98 35 02 81 02 26 34 E6 84 7E 5._.&4◥_
5CA0 20 28 81 23 26 10 35 02 59  (_#&.5.
5CA8 4A 17 F8 AF 25 24 10 83 E4 J. ｯ%$.■
5CB0 00 FF 22 1E 20 14 A6 82 9B ..". .ｦ_
5CB8 35 02 81 02 26 14 17 F7 02 5._.&..秒
5CC0 84 25 0F EC 84 17 F7 B8 EE _%.●_.秒ｸ
5CC8 1F 89 E7 8D FA 50 17 E5 62 .▎◤■ P.◣
5CD0 10 39 17 E5 0C 17 EB 7D D0 .9.◣..◢}
5CD8 16 FF 32 A6 8D FA 3F 36 E9 ..2ｦ■ ?6
5CE0 02 EC 8D FA 0E A3 8D FA AD .●■ .｣■
5CE8 08 C3 00 01 AE 8D FA 01 02 .ﾃ..ｮ■ .
5CF0 34 06 37 02 36 02 A7 80 D2 4.7.6.ｧ_
5CF8 35 06 83 00 01 26 F1 37 0D 5.■..&円7
----------------------------------------
SUM  F6 A2 39 D5 C2 AA 8F 56 F7
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5D00 02 39 8D 4A 8D 5D 25 42 63 .9■J■]%B
5D08 17 00 E8 17 01 4C 24 05 8C ..♠..L$.
5D10 17 EB 42 20 EF A6 8D FA 80 .♣B ＼ｦ■
5D18 06 81 2B 27 12 81 2D 27 C0 ._+´._-´
5D20 13 81 2A 27 14 17 01 A4 B5 ._*´...､
5D28 24 12 17 02 58 20 D5 17 B3 $...X ﾕ.
5D30 01 C9 20 08 17 01 D1 20 FB .ﾉ ...ﾑ
5D38 03 17 01 AB 17 01 D6 17 CB ...ｫ..ﾖ.
5D40 01 EC 17 02 0F 17 F5 D2 F3 .●....ﾒ
5D48 20 BA 17 F5 CD 39 17 F4 F7  ｺ.ﾍ9.日
5D50 FC 0D 0A 3E 43 41 4C 43 64 ..>CALC
5D58 55 4C 41 54 45 0D 0A 0D 9F ULATE...
5D60 0A FF 39 17 F4 E7 20 3F 93 ..9.日◤ ?
5D68 FF 17 E4 53 86 0C 17 F5 EB ..◢S■..
5D70 25 17 F4 D9 BC B7 3A FF B5 %.日ﾙｼｷ:.
5D78 86 04 17 F5 19 17 F4 CD 87 ■....日ﾍ
----------------------------------------
SUM  97 48 E5 45 DC 68 47 70 04
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5D80 20 48 45 58 3A 34 28 30 CB  HEX:4(0
5D88 30 30 30 2D 46 46 46 46 D5 000-FFFF
5D90 29 20 44 45 43 49 4D 41 EC ) DECIMA
5D98 4C 3A 27 23 27 2B 28 30 7A L:´#´+(0
5DA0 2D 36 35 35 33 35 29 20 7E -65535)
5DA8 42 49 4E 3A 27 25 27 2B B1 BIN:´%´+
5DB0 28 38 2C 31 36 42 49 54 D2 (8,16BIT
5DB8 2D 30 31 29 20 52 45 54 C2 -01) RET
5DC0 3A 71 75 69 74 FF 86 07 89 :quit.■.
5DC8 17 F4 CB EC 8D F9 20 17 7F .日ﾋ●■  .
5DD0 F4 9F 17 F6 CA 30 8D F9 20 日ﾞ.分ﾊ0■
5DD8 59 A6 80 81 FF 26 05 86 B0 Yｦ__.&.■
5DE0 01 17 EF 87 4D 26 06 17 1E ..＼■M&..
5DE8 E3 F7 1A 01 39 17 E3 F1 19 ┤秒..9.┤円
5DF0 1C FE 39 C6 04 31 8D FB D6 . 9ﾆ.1■
5DF8 02 34 04 30 8D F9 33 A6 C9 .4.0■ 3ｦ
----------------------------------------
SUM  29 A3 DD 00 7B 91 A2 20 77
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5E00 80 E6 A0 17 F7 33 4D 26 BA _◥..秒3M&
5E08 2A 35 04 5A 26 EB 86 2B 7F *5.Z&♣■+
5E10 A7 8D F9 0B A6 8D F9 1A 7E ｧ■ .ｦ■ .
5E18 A7 8D F9 04 4C 4C 30 8D 86 ｧ■ .LL0■
5E20 F9 10 30 86 86 23 A7 80 8F  .0■■#ｧ_
5E28 86 30 A7 84 86 02 A7 8D 9D ■0ｧ■■.ｧ■
5E30 F8 F1 39 35 04 A1 8D F8 81 円95.｡■
5E38 F9 26 06 6A 8D F8 F3 20 27 &.j■ 月
5E40 CD E6 A2 E7 8D F8 D8 4A E3 ﾍ◥｢◤■ ﾘJ
5E48 A7 8D F8 D4 A6 8D F8 E2 0D ｧ■ ﾔｦ■ ╪
5E50 A0 8D F8 CC 4A A7 8D F8 67 .■ ﾌJｧ■
5E58 CA 39 C6 02 31 8D F8 C0 41 ﾊ9ﾆ.1■ ﾀ
5E60 30 8D F8 CF 34 04 34 10 00 0■ ﾏ4.4.
5E68 34 20 A6 80 81 23 26 0A 4E 4 ｦ__#&.
5E70 A6 A4 4A 17 F6 E5 25 4F FA ｦ､J.分◣%O
5E78 20 26 81 25 26 12 A6 A4 6E &_%&.ｦ､
----------------------------------------
SUM  70 3C 6D 3D 2B 8C 44 0E 5F
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5E80 4A 81 08 27 04 81 10 26 B5 J_.´._.&
5E88 3E 17 F7 7E 25 39 20 10 58 >.秒~%9 .
5E90 A6 A4 81 04 26 31 E6 82 8E ｦ､_.&1◥■
5E98 17 F5 AA 25 2A 17 F5 CD DE .ﾈｪ%*.ﾍ
5EA0 35 20 ED 21 1F 20 C3 00 65 5 O!. ﾃ.
5EA8 03 1F 02 E6 8D F8 71 4F 4F ...◥■ qO
5EB0 ED 8D F8 45 35 06 E3 8D 62 O■ E5.┤■
5EB8 F8 3F C3 00 01 1F 01 35 50  ?ﾃ....5
5EC0 04 5A 26 A0 1C FE 39 35 AC .Z&.. 95
5EC8 34 1A 01 39 EC 8D F8 54 4D 4..9●■ T
5ED0 36 06 EC 8D F8 4B 36 06 34 6.●■ K6.
5ED8 17 F6 2D 25 09 37 06 ED 92 .分-%.7.O
5EE0 8D F8 43 37 06 39 39 EC 63 ■ C7.99●
5EE8 8D F8 39 36 06 EC 8D F8 6B ■ 96.●■
5EF0 30 36 06 17 F6 E6 ED 8D D9 06..分◥O■
5EF8 F8 2C 39 EC 8D F8 22 E3 D3  ,9●■ "┤
----------------------------------------
SUM  29 FE CF 15 F3 4F 65 66 18
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5F00 8D F8 21 ED 8D F8 1F 39 70 ■ !O■ .9
5F08 EC 8D F8 15 A3 8D F8 14 C2 ●■ .｣■ .
5F10 ED 8D F8 12 39 31 8D F8 73 O■ .91■
5F18 59 EC 8D F8 09 17 F4 F7 D5 Y●■ ..日秒
5F20 17 F3 2A 20 24 FF 31 8D 35 .月* $.1■
5F28 F8 48 17 F3 99 39 EC 8D 95  H.月┐9●■
5F30 F7 F4 30 8D F8 41 17 F6 EE 秒日0■ A.分
5F38 7A 34 14 17 F3 0F 20 23 1E z4..月. #
5F40 FF 35 14 34 04 A6 82 34 DC .5.4.ｦ_4
5F48 10 17 F3 41 35 10 35 04 D9 ..月A5.5.
5F50 5A 26 F0 39 EC 8D F7 CE E7 Z&X9●■秒ﾎ
5F58 30 8D F8 1B 17 F6 D5 86 38 0■ ..分ﾕ■
5F60 FF A7 85 C1 08 23 10 86 AD .ｧ■ﾁ.#.■
5F68 09 5C 30 85 E6 82 E7 01 6A .¥0■◥_◤.
5F70 4A 26 F9 86 5F A7 84 17 90 J& ■_ｧ■.
5F78 F2 D3 20 25 FF 31 8D F7 BE 年ﾓ %.1■秒
----------------------------------------
SUM  1C 5C E0 7D A2 0B 77 90 89
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
5F80 F6 17 F3 42 39 17 F2 C5 49 分.月B9.年ﾅ
5F88 20 64 69 76 69 73 69 6F 17  divisio
5F90 6E 20 62 79 20 30 20 65 3E n by 0 e
5F98 72 72 6F 72 0D 0A FF 39 14 rror...9
5FA0 17 00 8C 17 E3 23 17 00 D7 ..■.┤#..
5FA8 98 17 01 40 24 05 17 E8 18 r..@$..♠
5FB0 A4 20 F3 17 01 B3 AE 8D BD ､ 月..ｳｮ■
5FB8 F7 37 31 8D F9 9D E6 8D F5 秒71■ ､◥■
5FC0 F7 68 17 F6 BB 24 19 17 7B 秒h.分ｻ$..
5FC8 02 89 6D 8D F7 4B 27 06 F4 .▌m■秒K´.
5FD0 17 03 0E 17 02 C8 6C 8D 02 .....ﾈl■
5FD8 F7 4E 6A 8D F7 4B 27 3A DF 秒Nj■秒K´:
5FE0 A6 80 8C 00 00 27 06 AC 8B ｦ_■..´.ｬ
5FE8 8D F7 08 23 CD 6D 8D F7 6D ■秒.#ﾍm■秒
5FF0 38 27 27 17 F2 57 20 63 69 8´´.年W c
5FF8 61 6E 27 74 20 66 69 6E C7 an´t fin
----------------------------------------
SUM  13 C9 BC 73 5A 0F 2B 2C CB
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6000 64 FF 6D 8D F7 22 27 0D AA d.m■秒"´.
6008 17 F2 42 20 61 6E 79 20 D3 .年B any
6010 6D 6F 72 65 FF 17 F3 02 BE more..月.
6018 20 09 6D 8D F6 FB 26 03 3D  .m■分 &.
6020 17 02 7B 6F 8D F6 DB 6F D0 ..{o■分ﾛo
6028 8D F6 EE 17 F2 EC 39 17 B6 ■分/.年●9.
6030 F2 1B 0D 0A 3E 53 45 41 3B 年...>SEA
6038 52 43 48 0D 0A 0D 0A FF 0A RCH.....
6040 39 17 F2 09 20 54 61 72 92 9.年. Tar
6048 67 65 74 20 2F FF 30 8D 4B get /.0■
6050 F8 C9 E6 80 27 10 34 04 96  ﾉ◥_´.4.
6058 A6 80 34 10 17 F2 2E 35 D6 ｦ_4..年.5
6060 10 35 04 5A 26 F0 17 F1 C1 .5.Z&X.円
6068 E4 2F 20 3F FF 17 E1 4F B8 ◢/ ?..┼O
6070 86 0C 17 F2 21 17 F1 D5 99 ■..年!.円ﾕ
6078 54 41 52 47 45 54 3A FF 00 TARGET:.
----------------------------------------
SUM  FC 35 59 C7 2C AB 32 44 9E
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6080 86 04 17 F2 11 17 F1 C5 71 ■..年..円ﾅ
6088 20 48 45 58 3A 30 30 2D CC  HEX:00-
6090 46 46 28 31 2D 31 36 29 A2 FF(1-16)
6098 20 41 53 43 49 49 3A 27 EA  ASCII:´
60A0 22 27 2B 41 73 63 69 69 5D "´+Ascii
60A8 20 43 68 61 72 28 31 2D 24  Char(1-
60B0 31 36 29 20 52 45 54 3A D5 16) RET:
60B8 64 65 66 61 75 6C 74 FF E4 default.
60C0 86 07 17 F1 D1 EC 8D F6 D5 ■..円ﾑ●■分
60C8 26 17 F1 A5 17 F3 D0 A6 53 &.円･.月ﾐｦ
60D0 8D F6 5F 81 FF 26 05 86 13 ■分__.&.■
60D8 01 17 EC 8F 34 02 17 E1 C1 ..●┼4..├
60E0 00 35 02 4D 26 03 1A 01 C8 .5.M&...
60E8 39 1C FE 39 25 0C 30 8D 7A 9. 9%.0■
60F0 F6 40 86 01 A7 8D F6 01 E8 分@■.ｧ■分.
60F8 20 08 30 8D F8 1D 6F 8D F6  .0■ .o■
----------------------------------------
SUM  6C 9C 02 93 72 BD 1B 30 1F
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6100 F5 F7 31 8D F8 55 A6 80 1D 晴秒1■ Uヲ_
6108 A7 8D F6 1F 81 02 25 56 47 ァ■分.__.%V
6110 A6 80 81 22 27 26 A6 82 3E ヲ__"´&ヲ_
6118 A6 8D F6 0F 17 F3 26 25 8D ヲ■分..月&%
6120 45 A6 8D F6 06 44 25 3E 1B Eヲ■分.D%>
6128 A7 8D F5 FE 34 02 EC 81 CA ァ■時 4.●_
6130 17 F3 4D A7 A0 35 02 4A 1F .月Mァ.5.J
6138 26 F2 20 10 A6 8D F5 EB 5B &年 .ヲ■時♣
6140 4A A7 8D F5 E5 E6 80 E7 A5 Jァ■時◣◥_◤
6148 A0 4A 26 F9 A6 8D F5 DA 0B .J& ヲ■時レ
6150 81 10 23 06 86 10 A7 8D 84 _.#.■.ァ■
6158 F5 D0 6D 8D F5 9B 27 03 79 時ミm■時┘´.
6160 17 01 AE 1C FE 39 1A 01 34 ..ョ. 9..
6168 39 17 E9 21 17 E0 50 17 B8 9.♥!.═P.
6170 E6 D3 86 04 17 F1 1F 17 81 ◥モ■..円..
6178 F0 D3 20 6E 3A 66 69 6E C8 Xモ n:fin
------------------------------------------
SUM  97 38 0D B8 A3 06 D4 5F 70
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6180 64 20 63 6F 75 6E 74 28 D5 d count(
6188 31 2D 32 35 35 2C 27 2F 7C 1-255,´/
6190 27 3D 32 35 35 29 20 27 70 ´=255) ´
6198 2A 27 3A 76 65 72 69 66 A7 *´:verif
61A0 79 20 52 45 54 3A 63 6F 90 y RET:co
61A8 75 6E 74 3D 31 2C 6E 6F CE unt=1,no
61B0 20 76 65 72 69 66 79 FF B4  verify.
61B8 86 07 17 F0 D9 EC 8D F5 DB ■..Xル●■時
61C0 2E 17 F0 AD 17 F2 D8 30 F3 ..Xュ.年リ0
61C8 8D F5 67 A6 80 81 FF 26 B5 ■時gヲ__.&
61D0 05 86 01 17 EB 95 4D 26 96 .■..♣┤M&
61D8 10 86 01 A7 8D F5 4A 6F 79 .■.ァ■時Jo
61E0 8D F5 36 6F 8D F5 41 20 0A ■時6o■時A
61E8 59 34 02 C6 2A 17 F3 49 D2 Y4.ニ*.月I
61F0 4D 27 0C 86 01 A7 8D F5 30 M´.■.ァ■時
61F8 20 A7 8D F5 05 20 04 6F E1  ァ■時. .o
------------------------------------------
SUM  9D CB 6D F4 D7 BD 2E 6E F9
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6200 8D F5 16 35 02 34 02 C6 CB ■時.5.4.ニ
6208 2F 17 F3 2D 4D 27 0A 86 6A /.月-M´.■
6210 FF A7 8D F5 14 35 02 20 93 .ァ■時.5.
6218 29 35 02 E6 80 C1 2A 27 D8 )5.◥_チ*´
6220 04 E6 82 20 0B 4A 26 08 0F .◥_ .J&.
6228 86 01 A7 8D F4 FB 20 12 DC ■.ァ■日 .
6230 17 F3 28 25 15 10 83 00 FF .月(%..▃.
6238 FF 22 0F 5D 27 0C E7 8D 34 .".]´.◤■
6240 F4 E7 6F 8D F4 E2 17 DF A3 日◤o■日キ.°
6248 98 39 17 DF 94 17 E8 4B A5 ┌9.°┬.♠K
6250 16 FF 16 34 10 86 20 17 2C ...4.■ .
6258 F0 33 35 06 34 06 31 8D 56 X35.4.1■
6260 F5 10 17 F1 B2 17 F0 5E 24 時..円ィ.X^
6268 17 EF E2 20 20 46 69 6E 45 .\キ  Fin
6270 64 20 63 6F 75 6E 74 20 CD d count
6278 3D 20 FF E6 8D F4 A9 5C C8 = .◥■日ゥ¥
------------------------------------------
SUM  C3 75 24 78 BE F6 AE 50 86
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6280 4F 30 8D F4 ED 17 F3 2B 22 O0■日O.月+
6288 34 04 A6 82 34 10 17 EF AA 4.ヲ_4..\
6290 FC 35 10 35 04 5A 26 F0 EA  5.5.Z&X
6298 17 F0 7F 35 10 39 34 10 48 .X.5.94.
62A0 17 EF AA 20 57 72 69 74 76 .\ェ Writ
62A8 65 20 4D 6F 64 65 20 28 52 e Mode (
62B0 59 2F 45 4C 53 45 29 3F 19 Y/ELSE)?
62B8 FF 17 E8 1D 4D 27 13 35 D7 ..♠.M´.5
62C0 10 34 10 AF 8D F4 2E AF 61 .4.ッ■日.ッ
62C8 8D F4 39 6F 8D F4 33 17 F4 ■日9o■日3.
62D0 DE B4 6D 8D F4 43 27 06 F0 ゛Im■日C´.
62D8 86 01 A7 8D F4 24 35 10 18 ■.ァ■日$5.
62E0 39 34 10 AF 8D F4 0E AF 6A 94.ッ■日.ッ
62E8 8D F4 19 EC 8D F4 02 34 3D ■日.●■日.4
62F0 06 EC 8D F3 FE 34 06 AF 59 .●■月 4.ッ
62F8 8D F3 F6 AF 8D F3 F4 17 B0 ■月分ッ■月日.
------------------------------------------
SUM  C4 92 EF 4D 37 5B F0 AF C3
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6300 E1 65 35 06 ED 8D F3 EB D9 ├e5.O■月♣
6308 35 06 ED 8D F3 E3 35 10 D0 5.O■月┤5.
6310 39 30 8D F4 1D 31 8D F6 BB 90■日.1■分
6318 02 E6 8D F4 15 5C A6 80 00 .◥■日.¥ヲ_
6320 A7 A0 5A 26 F9 39 8D 13 99 ァ.Z& 9■.
6328 8D 25 17 EF ED 17 00 C4 80 ■%.\O..ト
6330 25 05 17 01 51 20 F6 17 C0 %...Q 分.
6338 EF E0 39 17 EF 0F 0D 0A 34 \═9.\...
6340 3E 52 45 47 49 53 54 45 51 >REGISTE
6348 52 0D 0A 0D 0A FF 39 17 CF R....9.
6350 EE FB 20 43 43 20 41 20 10 /  CC A
6358 20 42 20 20 44 50 20 58 AE  B  DP X
6360 20 20 20 20 59 20 20 20 39     Y
6368 20 55 20 20 20 20 50 43 88  U    PC
6370 20 20 20 53 50 0D 0A FF 19    SP...
6378 C6 04 30 8D F6 1D 34 04 D2 ニ.0■分.4.
------------------------------------------
SUM  5D 60 1C 7F D1 A8 87 A3 FB
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6380 86 20 34 10 17 EF 06 35 2B ■ 4..\.5
6388 10 A6 80 34 10 17 EE 2F AE .ヲ_4..//
6390 35 10 35 04 5A 26 E7 C6 AB 5.5.Z&◤ニ
6398 05 34 04 86 20 34 10 17 3E .4.■ 4..
63A0 EE EB 35 10 EC 81 34 10 CF /♣5.●_4.
63A8 31 8D F3 C6 17 F0 68 17 FD 1■月ニ.Xh.
63B0 EF 14 35 10 35 04 5A 26 01 \.5.5.Z&
63B8 E0 17 EF 5E 17 EE 8E 20 F7 ═.\^./■
63C0 45 46 48 49 4E 5A 56 43 5D EFHINZVC
63C8 0D 0A FF 17 01 61 17 EF 95 .....a.\
63D0 50 34 04 5A 86 21 17 EE 8E P4.Z■!./
63D8 98 EC 9D F5 CA 36 06 17 33 ┌●ヘ時ハ6..
63E0 ED DD 35 04 86 21 17 EE AF Oン5.■!./
63E8 88 37 06 1F 98 17 ED CF 4F ▎7..┌.Oマ
63F0 17 EF 27 39 17 EE 56 20 E1 .\´9./V
63F8 52 65 67 69 73 74 65 72 45 Register
------------------------------------------
SUM  D6 85 EA 86 37 6F B8 34 5D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6400 20 3F FF 17 DD B9 17 E4 06  ?..ンケ.◢
6408 3C 86 04 17 EE 88 17 EE 58 <■../▎./
6410 3C 20 72 65 67 69 73 74 EA < regist
6418 65 72 20 6E 61 6D 65 3A D2 er name:
6420 66 6F 72 20 43 68 61 6E E1 for Chan
6428 67 65 28 27 43 43 27 2D F5 ge(´CC´-
6430 27 53 50 27 29 20 52 45 D1 ´SP´) RE
6438 54 3A 71 75 69 74 FF 86 D6 T:quit.■
6440 07 17 EE 52 EC 8D F2 A7 70 ../R●■年ァ
6448 17 EE 26 17 F0 51 30 8D 40 ./&.XQ0■
6450 F2 E0 A6 80 81 FF 26 05 A3 年═ヲ__.&.
6458 86 01 17 E9 0E 4D 26 06 0E ■..♥.M&.
6460 17 DD 7E 1A 01 39 81 02 49 .ン~..9_.
6468 22 13 27 04 86 20 A7 01 AE ".´.■ ァ.
6470 17 00 D7 25 08 34 02 17 68 ..ラ%.4..
6478 DD 67 35 02 39 17 DD 61 09 ンg5.9.ンa
------------------------------------------
SUM  08 F5 72 FB DE 24 54 A0 60
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6480 17 E3 D2 16 FF 6E 34 02 85 .┤メ..n4.
6488 17 00 E3 17 F0 11 30 8D CF ..┤.X.0■
6490 F2 A0 A6 80 81 FF 26 05 63 年.ヲ__.&.
6498 86 01 17 E8 CE 4D 26 02 C9 ■..♠ホM&.
64A0 20 7C 81 02 25 7E 22 20 04  |_.%~"
64A8 35 04 34 04 C1 03 22 74 CB 5.4.チ."t
64B0 86 02 17 EF 90 25 6D EC 9C ■..\┴%m●
64B8 84 17 EF C4 30 8D F4 DB DA ▃.\ト0■日□
64C0 35 04 34 04 A7 85 20 56 13 5.4.ァ▃ V
64C8 81 04 26 33 35 04 34 04 4F _.&35.4.
64D0 C1 04 25 50 86 04 17 EF CA チ.%P■..\
64D8 6C 25 49 17 EF 8F 36 06 AB l%I.\┼6.
64E0 35 04 34 04 C1 04 26 08 64 5.4.チ.&.
64E8 37 06 ED 8D F4 AE 20 2E A7 7.O■日ョ .
64F0 C0 05 30 8D F4 A9 58 30 A7 タ.0■日ゥX0
64F8 85 37 06 ED 84 20 1F 81 F3 ▃7.O▃ ._
------------------------------------------
SUM  99 94 4C F7 62 95 B3 27 41
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6500 09 26 21 A6 80 81 25 26 42 .&!ヲ__%&
6508 1B 35 04 34 04 5D 26 14 23 .5.4.]&.
6510 86 08 17 F1 51 25 0D 17 30 ■..円Q%..
6518 F0 F0 E7 8D F4 7D 17 DC B8 ××◤■日}.ワ
6520 C0 35 02 39 17 DC BA 17 F4 タ5.9.ワコ.
6528 E3 2B 35 02 16 FF 57 86 37 ╕+5...W■
6530 20 17 ED 59 E6 8D F4 63 47  .OY◥■日c
6538 4F 30 8D F2 3A 17 F0 F4 33 O0■年:.×日
6540 86 FF A7 85 1F 12 17 ED E6 ■.ァ■...O
6548 7D 39 86 02 17 F1 45 4F DA }9■..円EO
6550 31 8D F3 AB 34 02 C6 02 5A 1■月ォ4.ニ.
6558 17 F1 25 25 0C E6 A1 35 1A .円%%.◥。5
6560 02 4C 81 0A 25 EE 1A 01 07 .L_.%/..
6568 39 35 02 1C FE 39 34 02 F9 95.. 94.
6570 86 20 17 ED 18 35 02 34 2D ■ .O.5.4
6578 02 30 8D F3 82 48 30 86 32 .0■月_H0■
-----------------------------------------
SUM  BA 81 40 3B 49 8E A7 51 85
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6580 A6 80 34 10 17 ED 06 35 A9 ヲ_4..O.5
6588 10 A6 84 17 EC FF 86 3D FF .ヲ▬.●.■=
6590 17 EC FA 30 8D F4 04 35 E7 .● 0■日.5
6598 02 34 02 81 03 22 12 30 20 .4._.".0
65A0 86 A6 84 17 EC 19 35 02 03 ■ヲ▬.●.5.
65A8 34 02 4D 26 1F 8D 80 20 F5 4.M&.■_
65B0 1B 81 04 22 04 EC 01 20 D3 ._.".●.
65B8 09 80 05 30 8D F3 E0 48 66 ._.0■月=H
65C0 EC 86 31 8D F1 AC 17 EE D2 ●■1■円ャ./
65C8 4E 17 EC FA 17 EC 7E 20 EC N.● .●~
65D0 3F FF 17 DB EA 17 E2 6D 80 ?..□◆.╪m
65D8 86 04 17 EC B9 35 02 34 B1 ■..●ケ5.4
65E0 02 81 03 22 30 17 EC 65 40 ._."0.●e
65E8 20 48 45 58 3A 32 28 30 C9  HEX:2(0
65F0 30 2D 46 46 29 FF 35 02 48 0-FF).5.
65F8 34 02 4D 26 2D 17 EC 4D 26 4.M&-.●M
-----------------------------------------
SUM  32 87 B4 9B 9A CA E6 F4 46
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6600 20 42 49 4E 3A 27 25 27 A6  BIN:'%'
6608 2B 28 38 62 69 74 2D 30 27 +(8bit-0
6610 31 29 FF 20 15 17 EC 35 C6 1). ..●5
6618 20 48 45 58 3A 34 28 30 CB  HEX:4(0
6620 30 30 30 2D 46 46 46 46 D5 000-FFFF
6628 29 FF 17 EC 20 20 52 45 02 )..●  RE
6630 54 3A 6E 6F 74 20 63 68 CA T:not ch
6638 61 6E 67 65 64 FF 86 07 8B anged.■.
6640 17 EC 53 EC 8D F0 A8 17 7E .●S●■×ィ.
6648 EC 27 35 02 39 17 00 83 1D ●'5.9..▬
6650 30 8D F0 D9 31 8D F2 75 AB 0■×ル1■年u
6658 17 00 86 25 F3 30 8D F3 65 ..■%月0■月
6660 4A A6 8D F0 CC 48 EC 86 F3 Jヲ■×フH●■
6668 10 83 FF FF 27 0A ED 8D 3C .▬..'.O■
6670 F0 BD 6C 8D F0 BB 20 20 91 ×ス1■×サ
6678 6D 8D F0 B5 27 0E ED 8D 4E m■×オ'.O■
-----------------------------------------
SUM  AB C5 C7 32 24 4A F4 78 43
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6680 F3 29 6F 8D F0 AB 30 8D 70 月)o■×ォ0■
6688 D9 92 1F 15 30 8D F0 9F EB ル┤..0■×ノ
6690 31 8D F2 44 8D 4B 25 F4 E5 1■年D■K%日
6698 17 00 95 A6 9D F0 90 A7 16 ..━ヲ、×┴ァ
66A0 8D F0 89 86 3F A7 9D F0 FF ■×┃■?ァ、×
66A8 86 10 EE 8D F2 F9 EC 8D 75 ■./■年 ●■
66B0 F0 7B 34 06 C6 03 30 8D 2B ×{4.ニ.0■
66B8 F2 EB 36 04 EC 83 34 06 C0 年♣6.●▬4.
66C0 37 04 5A 26 F5 C6 04 36 B0 7.Z&▚ニ.6
66C8 04 A6 82 34 02 37 04 5A F7 .ヲ_4.7.Z
66D0 26 F5 3B 17 EB 77 0D 0A E6 &▚;.♣w..
66D8 3E 47 4F 0D 0A 0D 0A FF 01 >GO.....
66E0 39 34 10 17 EB E0 35 10 A4 94..♣=5.
66E8 34 10 EC 84 31 8D F0 82 E4 4.●▬1■×_
66F0 17 ED 24 17 EB D0 17 EB FC .O$.♣ミ.♣
66F8 54 20 3F FF 17 ED A0 30 86 T ?..O.0
-----------------------------------------
SUM  80 E5 BB D8 37 44 BD 1D 4D
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6700 8D F0 2F A6 80 81 FF 26 78 ■×/ヲ__.&
6708 04 4F 17 E6 5E 4D 26 05 26 .O.◥^M&.
6710 35 10 1C FE 39 81 04 26 43 5.. 9_.&
6718 0F 17 ED 29 25 0A 17 ED 6F ..O)%..O
6720 4C 35 10 ED 84 1C FE 39 55 L5.O▬. 9
6728 17 E1 2A 35 10 1A 01 39 BB .├*5...9
6730 17 EB 1A 20 42 72 6B 20 7B .♣. Brk
6738 FF A6 8D EF F4 17 EA 7F 95 .ヲ■\日.◆.
6740 17 EB 0A 20 3D 20 FF EC 74 .♣. = .●
6748 8D EF E4 31 8D F0 23 17 48 ■\◢1■×#.
6750 EC C5 17 EB 71 6D 8D EF 0D ●ナ.♣qm■\
6758 D8 26 04 17 F3 95 39 6A 44 リ&..月━9j
6760 8D EF CE 17 F3 8D 6C 8D DA ■\ホ.月■l■
6768 EF C7 39 C6 02 17 D8 C9 6F \ヌ9ニ..リノ
6770 1F 40 C3 00 0C ED 8D F2 9A .@テ..O■年
6778 2E A6 8D EF AF A7 9D EF 32 .ヲ■\ッァ、\
-----------------------------------------
SUM  7F 6E 90 03 E4 62 EA E2 92
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6780 AE EC 8D EF AA ED 8D EF 29 ョ●■\ェO■\
6788 A4 ED 8D EF 64 17 EB 8A FD 、O■\d.♣
6790 17 FB BC 17 EB 84 10 EE 52 . シ.♣▬./
6798 8D EF 6F 6D 8D EF 92 27 8D ■\om■\┤'
67A0 06 30 8D FE B8 1F 15 30 DD .0■ ク..0
67A8 8D D8 71 1F 15 17 00 9B BC ■リq....┘
67B0 17 00 AF 17 00 DC 17 EC BC ..ッ..ワ.●
67B8 E6 17 DA 25 30 8D EF 72 1A ◥.レ%0■\r
67C0 A6 80 81 FF 26 04 4F 17 36 ヲ__.&.O.
67C8 E5 A1 4D 26 04 17 EB 4A 49 ◣。M&..♣J
67D0 39 34 02 A6 84 81 2A 26 6A 94.ヲ▬_*&
67D8 11 CC FF FF ED 8D F1 CB 11 .フ..O■円ヒ
67E0 6F 8D EF 4D 35 02 17 EB 71 o■\M5..♣
67E8 31 39 31 8D EF 89 6F 8D 9C 191■\┃o■
67F0 EF 07 35 02 34 02 86 04 ED \.5.4.■.
67F8 17 EC 4A 25 46 17 EC 6D 28 .●J%F.●m
-----------------------------------------
SUM  01 BC 3A 86 BC E3 82 F2 90
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6800 ED A1 6C 8D EE F3 A6 8D 9B O。l■/月ヲ■
6808 EE EF 81 07 25 04 35 02 C5 /\_.%.5.
6810 20 0D 1F 10 C3 00 05 1F 43  ...テ...
6818 01 35 02 80 05 2A D5 31 ED .5._.*ユ1
6820 8D EF 54 30 8D F1 84 A6 A8 ■\T0■円▬ヲ
6828 8D EE CE 34 02 EC A1 ED F9 ■/ホ4.●。O
6830 81 35 02 4A 26 F5 CC FF E8 _5.J&▚フ.
6838 FF ED 84 6F 8D EE F2 17 63 .O▬o■/年.
6840 EA D8 39 35 02 17 E0 0D 36 ◆リ95..=.
6848 16 FF 68 17 E9 FF 0D 0A 93 ..h.♥...
6850 3E 42 52 45 41 4B 20 50 13 >BREAK P
6858 4F 49 4E 54 0D 0A 0D 0A 68 OINT....
6860 FF 39 17 E9 E8 20 42 52 D4 .9.♥♠ BR
6868 4B 3A FF 30 8D F1 3C EC 5A K:.0■円<●
6870 81 34 10 10 83 FF FF 27 7D _4..▬..'
6878 13 31 8D EE F5 17 EB 97 4D .1■/▚.♣|
-----------------------------------------
SUM  01 0B AA 3D 43 73 1A F5 B8
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6880 17 EA 43 86 20 17 EA 05 F0 .◆C■ .◆.
6888 35 10 20 E3 35 10 17 EA 8E 5. ╕5..◆
6890 89 39 17 E9 B8 20 3F FF D8 ┃9.♥ク ?.
6898 17 D9 24 17 DF A7 86 04 3B .ル$.°ァ■.
68A0 17 E9 F3 17 E9 A7 20 61 1B .♥月.♥ァ a
68A8 64 64 72 65 73 73 3A 48 07 ddress:H
68B0 65 78 34 28 37 20 70 6F 6F ex4(7 po
68B8 69 6E 74 73 2C 74 65 72 35 ints,ter
68C0 6D 69 6E 61 74 65 64 20 02 minated
68C8 73 70 61 63 65 29 20 27 7C space) '
68D0 2A 27 3A 63 6C 65 61 72 92 *':clear
68D8 20 62 72 6B 20 70 6F 69 C7  brk poi
68E0 6E 74 20 52 45 54 3A 71 98 nt RET:q
68E8 75 69 74 FF 86 07 17 E9 DE uit.■..♥
68F0 A5 EC 8D ED FA 17 E9 79 7E ・●■O .♥y
68F8 39 00 00 00 00 00 00 00 39 9.......
-----------------------------------------
SUM  20 6A 47 50 D5 71 83 71 5B
```

ダンプリスト2 FM-7裏RAM利用プログラム〔URA〕

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6D00 34 7F 17 00 E9 2A 20 55 52 4...♥* U
6D08 52 41 2D 52 41 4D 20 4D 0D RA-RAM M
6D10 4F 44 45 20 3F 28 59 2F E7 ODE ?(Y/
6D18 45 4C 53 45 29 FF 17 00 68 ELSE)...
6D20 F1 4D 26 13 17 00 C7 20 75 ■M&...ヌ
6D28 20 63 61 6E 63 65 6C 65 EB  cancele
6D30 64 0D 0A FF 35 7F 39 BE 25 d...5.9セ
6D38 FB FA AF 8D 00 7E 31 8D 6D   ッ■.~1■
6D40 C2 BE CC FB FF A3 8D 00 76 ツセフ .｣■.
6D48 73 ED 8D 00 71 8D 71 BD 19 s■■.q■qス
6D50 EC 05 B7 FD 0F 10 AE 8D FF ●.ｷ  ..ｮ■
6D58 00 62 10 BF FB FA 30 8D E3 .b.ｿ  0■
6D60 C2 9E EC 8D 00 58 8D 58 16 ツ■●■.X■X
6D68 CC 29 00 8E 40 00 10 8E 61 フ).■@..■
6D70 C0 00 8D 4C B6 FD 0F 8D E8 タ.■Lカ .■
6D78 75 2A 20 55 52 41 2D 52 26 u* URA-R
----------------------------------------
SUM  6E 0A D5 37 03 D0 02 3D 96
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6D80 41 4D 20 4F 4B 0D 0A 2A 89 AM OK..*
6D88 20 4C 6F 61 64 20 64 65 89  Load de
6D90 62 75 67 20 70 72 6F 67 16 bug prog
6D98 72 61 6D 28 31 30 30 30 29 ram(1000
6DA0 2D 36 45 46 46 29 20 61 DE -6EFF) a
6DA8 6E 64 20 45 78 65 63 20 97 nd Exec
6DB0 26 48 36 46 30 30 0D 0A 61 &H6F00..
6DB8 FF 35 7F 39 00 00 00 00 EC .5.9....
6DC0 34 06 A6 80 A7 A0 35 06 E2 4.ｦ■ｧ.5.
6DC8 83 00 01 26 F3 39 B6 FD 89 ■..&■9カ
6DD0 05 2B FB 86 80 B7 FD 05 EA .+ ■■ｷ .
6DD8 B6 FD 05 2A FB 39 4F B7 1C カ .* 9Oｷ
```

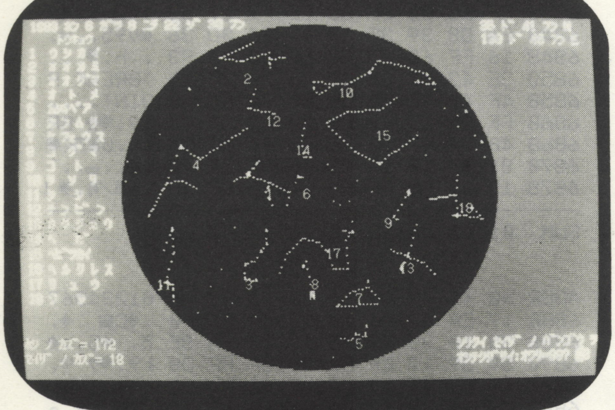
```
6DE0 FD 05 39 B6 FC 80 8A 80 77  .9カ ■■
6DE8 B7 FC 80 8D F1 39 35 10 2F ｷ ■■■95.
6DF0 8D DC 10 8E FC 81 6F A0 93 ■ﾜ.■ ■o.
6DF8 86 03 A7 A1 5F A6 80 81 D7 ■.ｧ｡_ｦ■■
----------------------------------------
SUM  2E 94 94 CA 9B 36 82 21 94
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6E00 FF 27 05 A7 A0 5C 20 F5 E3 .'.ｧ.¥ ■
6E08 10 8E FC 83 E7 A4 8D CE 03 .■ ■■､■ﾎ
6E10 1F 15 8D 1D 4D 27 FB C1 0E ..■.M' ﾁ
6E18 59 26 02 20 04 C1 79 26 05 Y&. .ﾁy&
6E20 09 8D CB 59 0D 0A FF 86 56 .■ﾋY...■
6E28 01 39 8D C2 0D 0A FF 4F EE .9■ﾂ...O
6E30 39 8D 9B 8E FC 82 86 29 1C 9■■■ ■■)
6E38 A7 80 6F 84 8D A0 8D 8E 62 ｧ■o■■.■■
6E40 8E FC 83 E6 80 A6 84 34 D1 ■ ■■■ｦ■4
6E48 02 8D 98 35 02 39 00 00 97 .■■5.9..
6E50 00 00 00 00 00 00 00 00 00 ........
6E58 00 00 00 00 00 00 00 00 00 ........
6E60 00 00 00 00 00 00 00 00 00 ........
6E68 00 00 00 00 00 00 00 00 00 ........
6E70 00 00 00 00 00 00 00 00 00 ........
6E78 00 00 00 00 00 00 00 00 00 ........
----------------------------------------
SUM  01 4C 0D AF FD FD B6 6A 23
```

6E80～6EF8はすべて00

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6F00 7F FD 0F BD C0 00 B6 FD BB . .ｽﾀ.ｶ
6F08 0F 39 00 00 00 00 00 00 48 .9......
```


星空への招待 パソコンプラネタリウム シリーズ



■適用機種

FM-7,8,11／PC-8801,9801

■主要機能

1. 世界のあらゆる場所、任意時刻の星空表示
2. 日出没時刻、白夜表示
3. 星の等級、色に応じた表示、星の名前表示
4. 指定星座内の星を点線で連結
5. 指定エリアの拡大表示

パソコンプラネタリウムI……機能1～3 ¥3,500
パソコンプラネタリウムII……機能1～5 ¥5,500
（※フロッピーは1,000円高。）

JIS 日本情報システム（株）
TEL. 03-364-0571
〒160 東京都新宿区高田馬場4-11-8 大都会ビル

6809機械語(マシン・ランゲージ)マスタ'84

第1回 マシン語ってなにものだ

鶴岡哲明

機械語はいま非常にメジャになっていてマイコンを始めてしばらくたつと，どうしても無視できない響きを伴ってきます。この機械語とやらを，年度変わりで気分一新したところでチャレンジしてみようじゃありませんか。ということで，今月から6809（もちろんFMシリーズに使われているCPUです）の機械語入門の連載を開始します。初心者向けでスタート（もちろん終わりの内容レベルは高くなります）しますので，ためらわず読み進めてほしいと思います。

序²章 なぜ機械語(マシン・ランゲージ)？

機械語を知らない人は，必然的に多くの誤解，偏見を持っているのではないかと思います。そこで，この章では機械語を始める準備段階の準備段階（だから序の序＝序²です）として，機械語がどんなものか考察し，本当に学ぶべきものか考えることにします。

機械語は高級か!?

機械語と聞くと，高級そうな響きがあります。でもBASICのほうが高級言語であることは事実です。このギャップはどのへんにあるのでしょうか。

BASICが高級と言われるのは，その機能からです。つまり，人間に身近なものとなるようにいろいろ工夫されています。単純な英語と数値を並べるだけでプログラムを作れます。簡単な命令でキーボードからの入力ができます。出力も同じです。これらはINPUT命令やPRINT命令で容易に実現できます。

こういったことは，インタプリタという機械語で作られたプログラムが人間にわかりやすく書かれたプログラムをみて，その都度何を実行すべきか判断し実行する，といった処理をやってくれるからです。つまり，コンピュータがより高度な処理をするから高級なのです。

一方，機械語のほうは，コンピュータが実行しやすい単なるコードにすぎません。そのために人間にとっては一体何がどうなっているのかわかりません。そのため，それぞれの命令コードをアルファベットで表すなどの処理が必要となります。しかし，これは逆に機械には都合がよいわけで，BASICのように判断などの手間がなく，実行速度が速いのです。この処理速度が機械語神話を生み出している主な要因ですし，機械語は難しいということが，その神話をさらに神話化しているともいえます。このへんのイメージが，機械語は高尚すなわち人間側から見ての高級さのイメージとなっているのです。

しかし，機械語の方が原始的(？)であることは，BASICがパソコンのように画面やキーボードの付いているシステムを前提としているのに対し，機械語はCPUに少しメモリがあれば実行可能なことでもわかると思います。今はBASICをやって機械語をやるというパターンが多いのですが，昔はとてもマイコンと思えないワンボードマイコンという1枚の基板にスイッチと

図1 BASICと機械語プログラム

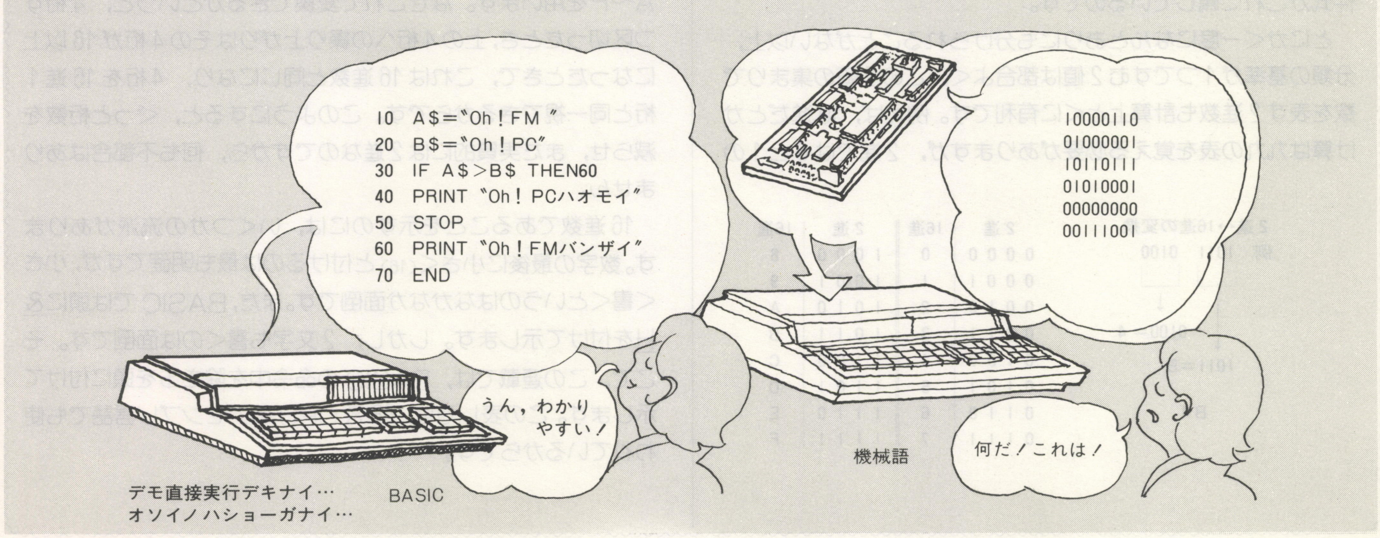
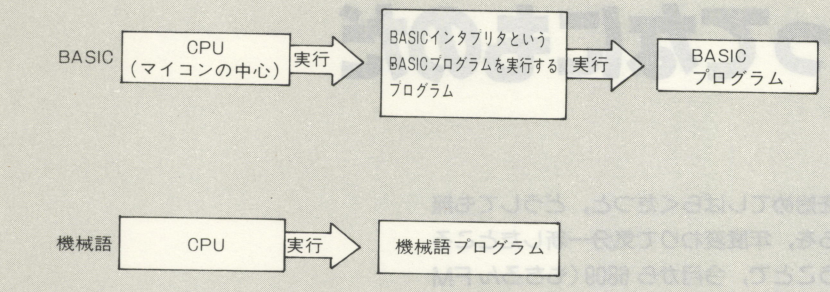


図2 BASICと機械語の処理



LED(電卓の表示器みたいなもの)とICがポロポロと載っているもので，機械語をやるのが第一歩だったのです。こんなシステムでBASICなんて使えません。

機械語は難しい？

機械語は難しいとよく言われます。しかし，本当に難しいのでしょうか。

実は機械語を学ぶとき覚えることは，そう多くないのです。機械語はその名のとおり，機械すなわちハードウェアと密接な関係を持っているので，これを少し勉強する必要はありますが，6809 CPUの命令は59種，これにアドレッシングモードを10種ちょっと，覚えればよいのです。このうち，命令をもっと大まかに分ければ半分くらいになってしまいます。各命令のバリエーションも単純で，BASICのように1つの命令のバリエーションがいろいろあって，そのたびにマニュアルをひっくり返す必要もありません。1つの命令が単純なのです。だから，すぐ覚えられます。機械語を知っているというレベルまではすぐ行きつくことができます。

しかし，実際にプログラムを作る段階になるとどうでしょう。機械語にはINPUT命令やPRINT命令などありません。BASICで簡単にできることでもINPUTのサブルーチン，PRINTのサブルーチンという具合にプログラムをわざわざ作ってやらなければなりません。システムコールといって，前もってROMに入っているプ

2進の話・16進の話

コンピュータは2進が基準なのはみなさん知っていると思います。しかし，なぜ2進なのでしょう。実は2進数はほかのn進法(5進数，10進数)とは持つ意味がずいぶん異なるのです。というより，コンピュータは2値(2とおりの値をとるもの)の論理で構成され，これで数値を表そうとすると2進法がぴったりくるからです。そして，この2値の論理はYesかNoかなのでなにかと便利で，種々の演算も基本演算の組み合わせでできます。こう書くとずいぶん難しい話になったように思えるかもしれませんが，BASICでもよく使われています。IF文の条件式がこれに属しているのです。

とにかく一度になんとおりにも分けられることがない以上，分類の基準が1つですむ2値は都合よく，この2値の集まりで数を表す2進数も計算上とくに有利です。例えば，10進だとかけ算は九九の表を覚える必要がありますが，2進では0と1のみ，普通覚えるのは2の段からなので，2以上の数字がないから九九の表は不要です。おまけに1桁同士なら繰り上がりもありません。

2進↔16進の変換

例 1011 0100

0100＝4

1011＝B

B4

2進	16進	2進	16進
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

このように，2進数というのはn進数の中でも非常に処理しやすいのです。しかし，この2進数にも欠点はあります。それは桁数が非常に多くなることです。そのため，16進数で表わすという方法が用いられます。

2進と16進の変換は実に単純で，小数点の位置から4桁ごとに区切り，0000〜1111の2進4桁(これで$2\times2\times2\times2=16$とおりのパターンがある)を16進数1桁に対応させればよいのです。数字は10個しかないので，16進では10〜15を表すのにA〜Fを用います。なぜこれで変換できるかというと，4桁ずつ区切ったとき，上の4桁への繰り上がりはその4桁が16以上になったときで，これは16進数と同じになり，4桁を16進1桁と同一視できるからです。このようにすると，ぐっと桁数を減らせ，また実質的には2進なのですから，何も不都合はありません。

16進数であることを示すのには，いくつかの流派があります。数字の最後に小さく(16)と付けるのは最も明確ですが，小さく書くというのはなかなか面倒です。また，BASICでは頭に&Hを付けて示します。しかし，2文字も書くのは面倒です。そこで，この連載では，BASICの命令中を除き$を頭に付けて示します。この表し方は，68系CPUのアセンブリ言語でも使われているからです。

ログラムを呼び出すことができる場合もありますが，BASICより手間がかかることに変わりありません。

数値計算も同様です。BASICでは数式どおりに書けばよかった計算が，機械語では1つずつやることになります。当然各演算もROMコール(BASICインタプリタには当然演算ルーチンが用意されているので，それをサブルーチンコールする）するか，自分で演算ルーチンを組む（考えるのもイヤなくらい実際に行う内容とプログラムでやらなければいけない内容にギャップがあります）必要があるのです。

要するにプログラム作りが面倒なのです。

それだけではありません。デバッグも面倒です。BASICのように親切なエラーメッセージが出ません。BASICでは変なことをするとエラーメッセージを出して止まってくれることが多いのですが，機械語では暴走が待っていることがほとんどです。暴走したら全焼の火事といっしょで，何が火元かわかるものではありません。

というわけで，結論を申しあげましょう。「理屈」は簡単，でもプログラムを作るのは……。

結局機械語のメリットは？

プログラミングが面倒なら，速さも夢に終わると言いたいところですが，機械語にはまだ長所があります。原始的ゆえにキメ細かな処理ができるのです。BASICが有利なのは型にはまったデータ（決められた型の数値，文字列）だけで，これに当てはまらないものを処理すると，手間は機械語並み，場合によってはそれ以上になることもあるのです。とくにハードウェアを新たに付けたりして，そのコントロールプログラムを作ろうとすると，これがよく当てはまります。場合によってはスピードを要求するので，BASICではプログラミングできないこともあり得ます。こういったケースでは機械語を使わざるを得ません。

またユーティリティ（実用）ソフトでメモリ上のBASICプログラムを調べるもの（例えばOh! FM Vol. 5 Q-Lab.のプログラミングツール)は，どうしてもBASICを使うわけにはいきません。こういったケースでは機械語は有益です。

また機械語はプログラムが短くなるという特徴があります。もちろん，ここでいう長さとは，命令の個数ではなく，実際にプログラムを入れるのに必要なメモリです。

機械語は速度以外にも上記のようなメリットがあり，実際にはBASICと短所を補いながら用いると，開発の手間とソフトの機能がよくなります。ということで，機械語は，知っていて損はないものといえると思います。

表1 BASICと機械語

	BASIC	機械語
実行速度	遅い！	めちゃ速い
命令	まあまあ覚えやすい 個々の命令を細かく覚えるのが面倒	まあまあ覚えやすい 個々の命令もバリエーションもワンパターン
プログラム作り	楽	なかなか面倒
デバッグ	楽	パニック
向いている処理	整数型・実数型・文字列・配列といったもので扱えるデータには向いているがこれ以外は狂気的	結局何でもやれるがプログラム作りは面倒。データが1バイトで入れば楽
構造化プログラミング	ムチャ	機能的にはやってできないことはない。工夫次第
走るシステム	キーボード，ディスプレイがないと。BASICインタプリタも大きいプログラムなのでメモリも多く必要	自作のマイコンでもすぐ走る

序章 機械語(マシン・ランゲージ)の基礎知識

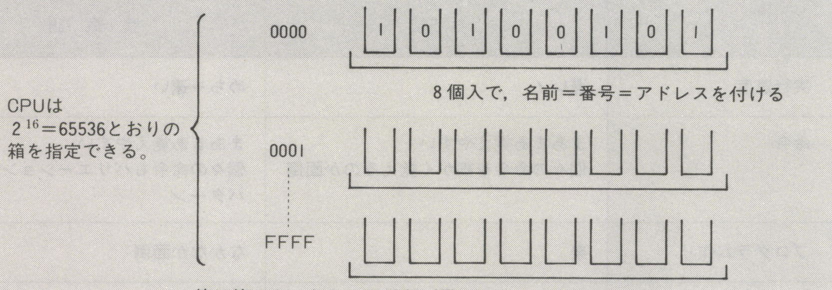
[1] CPUってなんだ

CPUがなんなのか，あーだこーだ能書きがありますが，やはり「機械語を解釈し実行するところ」というのが初心者にはよい説明（実際にはCPUあっての機械語ですが……）ではないかと思います。ついでに付け加えると，CPUはCentral Processing Unitの略で，日本語では中央演算処理装置などと訳されます。

CPUは，機械語の命令をメモリ（記憶装置）から読み出し，解釈し，その内容に応じた計算などの処理をし，必要ならばメモリからの読み出し，書き込みも行います。

機械語はCPUによって異なります。だから，これから学ぶ機械語は例えばPCシリーズやMZシリーズでは使えません。しかし，FMシリーズで6809 CPUを使っているかぎり，機械語は同一です。しかし，

図3



☆箱は箱でも，なかには役割が異なるものもあります

FMシリーズでも他の構成が異なるために同じプログラムが走らないことがあります。このことから，FM-7とFM-11では機械語が違うなどという表現をすることがありますが，正しくは機械語で処理する内容を違えなければならないことがあるということです。CPUは同じ動作をしても，ROMの内容が違っていたりI/Oの働きが違ったりすると，実際のパソコンシステムとしては同じ動作が得られないからです。

2 メモリ，アドレス，データ

メモリはその名のとおり何かをおぼえるところです。コンピュータでこの言葉を使う場合，“1”か“0”かの2つのうちの1つを表すデータを最小単位として記憶します。この最小単位ってもちろんビット(bit)です。

さて，実際にメモリを使うとなると，メモリのどこを読み，どこに記憶するのかを指定する必要があります。手帳を例にとると，何ページか，または何日の欄かというように指定すると思います。コンピュータでは通常1か0かのデータがいくつか入る箱（8ビットCPUなら8個）に通し番号を付け，この箱単位に通し番号で指定し，読み/書きを行うのです。この通し番号はアドレスと呼ばれ，6809では2^{16}個の番号（0～65535，16進で0000～FFFF）が付けられます。これは，16ビットでアドレスを示すからです。

3 プログラムとデータ

BASICを使っているとき，プログラムがどのようにメモリに入っているか，また，変数がどうメモリに格納されるか考えることはめったにありません。これはBASICがメモリの使い方を決め，管理しているからです。

しかし，機械語ではこういった管理は行われません。ここがプログラムエリア，ここが変数エリアということは，プログラムする人が決めるのです。

例えば，データエリアをプログラムエリアと勘違いしてプログラムしたとすると，CPUはそのプログラムの処理を忠実に守り，データエリアを機械語プログラムとして実行してしまいます。また，1つの命令の途中に飛び込んでも同様で，飛び込んでしまった以上CPUは命令の先頭と受け取って処理します。このへんが機械語の暴走しやすい要因ともいえます。

このように，プログラムもデータも，同じようにメモリに格納されているんだということは知っておいてください。

4 機械語を試そう！

前置きが長くなりましたが，機械語がどんな形なのか具体的に見てみましょう。

6809は8ビットCPUですから，命令は8ビット単位です。**プログラム1**は，3つの命令から成っているプログラムですが，それぞれの命令の長さは，見てのとおりまちまちです。命令1は2バイト命令，命令2は3バイト命令，命令3は1バイト命令です。

ビットとバイト

2進数1桁では2とおりの事柄が表せます。2通りの一方を示せると言い換えてもよいでしょう。つまり，2進数1桁では2とおりの一方であるという情報を表せます。これが1ビット（bit=binary digit）の情報量です。2ビットでは2とおり×2とおり＝4とおりの情報を表せます。8ビットでは256とおりです。つまり，2進数8桁は0～255の256とおりの数を表せます。

しかしビットという単位は2とおりが最小単位なので細かすぎます。例えば　英語で使う文字はアルファベット26文字，小文字と大文字で計52文字，数字や記号を入れると90文字ほどあります。そうすると，6ビット＝64とおりでは不足で，7ビット（＝128とおり）あったほうがよいことになります。しかし，7というのは2進には中途半端なので，思い切って8ビットにしたくなります。というような感じで，8ビット単位で情報を扱うと都合がよく，CPUも8ビット単位で扱うものが多くあります（8ビットCPU）。こうなると8ビットを1つの単位として名前を付けたくなり，これが1バイトとなりました。

プログラム1

アドレス	データ	
5000	86	命令1
5001	41	
5002	B7	命令2
5003	51	
5004	00	
5005	39	命令3

さて，このプログラムは，$5100番地に$41を書き込む(=POKE&H5100, &H41)プログラムです。

試しに実行してみましょう。

プログラムの入れ方ですが，「BIOSの使い方」でも説明されていましたし，たいていの人は機械語を打ち込んだことがあると思いますが，念のため説明します。

まず，

MON↵(ここで＿は入力すべき文字を示します。)

と入力します。すると

＊■

と出ます。これでモニタモードになりました。ここから先は数値はすべて16進になります。ここでまず，

＊M5ØØØ↵

と打ちます。Mはメモリに書き込むコマンドで，5000は$5000番地から書き込みをスタートさせることを意味します。すると，

5ØØØ　ØØ-■

と表示されます。左からの4文字はアドレスを示します(16進)。次の2文字は現在そのアドレスに入っているデータの内容を示します。このとき↵を押すと，メモリの内容は変わらず，次のアドレスへ移りますが，プログラムを打ち込むのですから，

5ØØØ　ØØ-86↵

と打ちます。これで$5000番地には$86が入りました。これでアドレスは1つ進み，

5ØØ1　ØØ-■

と表示されますので，以下同様に**プログラム1**を打ち込んでください。$5005番地の$39を入れ終えたら，

5ØØ6　ØØ-P↵

と入力し，むりやりエラーを起こしてMコマンドを終了させます。

＊D5ØØØ↵

と入力して，左上からの6つのデータが**プログラム1**と合っているか確認してください。Dコマンドは8バイトずつ横に並び，必ずこれが8行表示されます。余分なところは無視してください。たった6バイトで入力ミスなどないと思いますが，あればMコマンドで再入力します。もちろん間違った個所だけ直せばよいわけです。

モニタモードはBREAK(STOP)キーで抜け，続いてこのプログラムを試してみます。

機械語プログラムは，EXEC命令で実行します。今の場合，プログラム開始アドレスが$5000番地なので，

EXEC &H5ØØØ↵

と入力すれば機械語プログラムを「あっ」とも言えぬ間に終えて，

Ready

となります。

さて，$5100番地は$41が書き込まれたでしょうか。モニタのMコマンドかDコマンドで調べてみてください。また，$5100番地に例えば00を入れ，もう1度先ほどのプログラムを実行し，また$41になったか調べてみるのもよいでしょう。

5 プログラム1の正体

では**プログラム1**の種あかしをしてみましょう。

$5100番地に書き込む$41というデータを直接特定のアドレス(今の場合$5100番地)に書き込む命令は，6809 CPUにはありません。そこでCPU内にあるレジスタ(CPU内にある，数をおぼえておくところ。とりあえずBASICの変数みたいなものが数個CPU内にあるとだけ覚えておいてください)にこの数を入れるのです。そして，レジスタの内容を今度は$5100番地に入れてやれば，目的の操作が完了します。

命令1はレジスタに$41を入れるという命令で，命令2はレジスタの内容を$5100番地に入れる命令です。なお，使っているレジスタはAレジスタといって，8

PEEKとPOKE

メモリにはそれぞれ個々にアドレスが付けられているのはわかったと思います。そこで，BASICで直接アドレスを指定し，メモリの内容を書き換えるPOKE命令，特定アドレスのメモリの内容を読み出すPEEK関数について説明しましょう。

これらのBASICの命令は機械語プログラムを扱う際，非常に有効です。

まずPOKE命令ですが，マニュアルなどにもあるとおり，

POKE　address, data

で，アドレスaddress番地の内容をdataにします。

POKE &H51ØØ, &H41

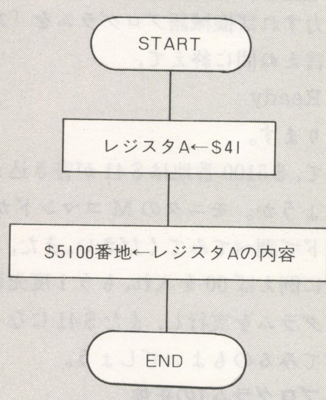
とやれば，$5100番地に$41が書き込まれます。

また，PEEKは読み出しで，PEEK(address)でaddress番地の内容を読み出します。例えば，

? PEEK(&H51ØØ)

とすれば，$5100番地の内容がCRTに表示されます。

図4



ビットの長さを持ちます。また，命令3はプログラム終了のしるしと考えてください。説明はもっと先で詳しくしたいと思います。

図4にフローチャートを示したので，参考にしてください。

⑥ ニモニック

機械語の命令は一目みただけでは何をやっているかわからない，ということが今の例でわかったと思います。そこで，プログラムを作りやすいよう，ニモニックで命令を書き表します。

例えば，命令1はメモリの内容をCPU内のレジスタに持ってくる命令で，この操作をロード(Load)といい，ニモニックでは，

　LD

と書き表します。実際には，レジスタ名もニモニックに含め，

　LDA

と書きます。また，何をレジスタにロードするのかを指定する必要があるので，オペランド（操作対象）として付け足します。命令1は$41という数値だから，

　LDA #$41

と記します。#は直接数値を指定することを示す記号として使います。

同様に命令2も表してみましょう。

こちらはレジスタの内容をメモリに入れるストア(Store)命令，対象レジスタはA，操作対象は，今度は直接数値ではなくアドレスで示して，

　STA $510Ø

となります。今度は#がない点に注意してください。

ついでに最後の命令は，

　RTS

です。これは操作対象の指定不要な命令なので，これだけです。

⑦ アセンブリ言語とアセンブラ

このように，ニモニックで命令を表すとぐっとわかりやすくなります。この形式でプログラムは書き表せますが，機械語の数値（命令コード）がずらずら並んだものとは1対1で対応していても，異なった言語なので，この言語をアセンブリ言語といいます。

アセンブリ言語は，機械語に直さなくてはCPUが実行できません。この変換作業をアセンブルと呼び，この作業をやってくれるプログラムがアセンブラです。また，手作業でやることはハンドアセンブルといいます。

なお，アセンブリ言語とアセンブラを混同して，「アセンブラで書いたプログラム」などというケースが多く見受けられることを覚えておいてください。でも，このほうが言いやすいですね……。

さて，機械語とアセンブリ言語の対応を図5に示しました。なお，アセンブリ言語でORGはアセンブルするプログラムの先頭アドレスを示すための命令，ENDはアセンブリ言語のプログラム終了を示す命令で，ともに疑似命令と呼び，機械語にはなりません。

◇　　◇　　◇

今月はここまでです。次回は今回ふれたロード，ストア命令から始め，基本的な命令をどんどんやっていきたいと思います。

図5

```
5000 86
5001 41
5002 B7
5003 51
5004 00
5005 39
```

(a) 機械語

```
NAM   TEST
ORG   $5000
LDA   #$41
STA   $5100
RTS
END
```

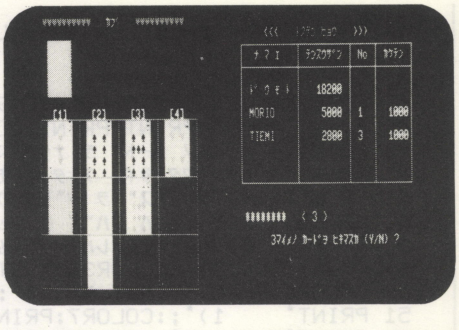
(b) アセンブリ言語
ソースリスト

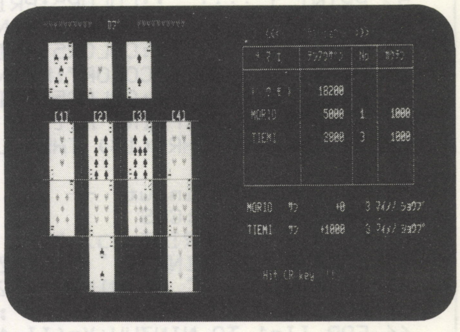
```
00110  5000                    ORG   $5000
00120  5000 86    41           LDA   #$41
00130  5002 B7    5100         STA   $5100
00140  5005 39                 RTS
00150                          END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=5000
PROGRAM END   ADDR=5005
```

(c) アセンブルリスト

for FM-7,8,11

ゲームリスト
パルコム「おいちょかぶ」





「おっさん,またバクチしてん？」「あほ！これバクチやないでえ。お前知らんやろ,今はやりのコンピュータゲームやねん」

そう，みなさまおなじみの〝おいちょかぶゲーム〞です。パソコンが胴元で，３人まで遊ぶことができます。もちろん親のクッピン，子のシッピン，アラシ（スリーカード）の役もちゃんと考えています。

あそび方

まず，プレイする人数と名前を聞いてきますので，それぞれ入力します。次に画面には胴元の札１枚と，掛け札４枚が出てきます。子（プレイヤ）は掛け札４枚のうち好きなものに持ち金以内を掛け，子が掛け終わった後で胴元が次の札を配ります。そこで２枚の札の合計をみて，下１桁が９に近いときは有利ですので〝もう１枚？〞と聞いてきたときに〝N〞を押します。まったくだめなときは，もう１枚もらうことができます。とくに２枚めで１と４か，４と１の組み合せになったときは，子のシッピンの役がつきますので当然〝N〞を押してください。また，３枚のカードがすべて同じ数字になったときはアラシの役がつきます。いちばん強いのはこのアラシで，掛け点の３倍が返ってきます。次に親のクッピン(親が２枚で９と１をそろえたとき)，子のシッピン，カブ(下１桁が９)，おいちょ(8)，ナキ(7)，6，5，4，3，にたこ(2)，いんけつ(1)，ブタ(0)の順番です。

親と子が同じ強さのときと，親がブタになったときは，掛け点は返ってきます。親がアラシのときは掛け点の３倍とられ，親のクッピンに負けた（子がアラシ以外のとき）場合は２倍，その他で負けたときは掛け点だけとられてしまいます。子がブタになったときは，親がブタになる以外に救われないということです。

単純なようで，掛け点の掛け方，４枚の札に掛ける場所など，駆け引きがとても難しいゲームです。すぐにハダカにされてしまいます。がんばって胴元をたおしてください。なおFM-11の場合，プログラムの関係上，♥と◆が黒で表示されます。ご了承ください。

＊　＊

「おいちょかぶ」は㈱パルコム（大阪市北区梅田１-１-３-900　大阪駅前第３ビル９Ｆ　☎06(341)3422)よりFM-7，8用として発売されています。製品に対するお問い合せは上記へ，ただしバグ情報などはOh!FM編集部へお願いします。

おいちょかぶリスト

©1983 PALCOM

（FM-7,8の方は□で囲まれた部分を訂正してください。またFM-11の方はSCREEN0を実行してください）

```
1 '****************************************************************
2 '*                                                              *
3 '*                  ｵｲﾁｮ ｶﾌﾞ   FM-8 version                     *
4 '*                                                              *
5 '*                        1983. 20. MAY   -- PALCOM --          *
6 '*                                                              *
7 '*              by  T.Imoto          modified by H.Ushida       *
8 '*                                                              *
9 '****************************************************************
10 'save"CAS0:ｵｲﾁｮｶﾌﾞ"
11 CLOSE
15 WIDTH 40,25:CONSOLE 0,25,0,0:COLOR 7,1:CLS
20 DIM SW(40),C1(40),NAM$(2),TEN(3),CARD(3,4),AAX(4),ZXX(2,1),BBX(4),CCX(2,1)
23 DIM OICH%(64)
25 RANDOMIZE( VAL(RIGHT$(TIME$,2))+60*VAL(MID$(TIME$,4,2)))
30 GOSUB 10000
31 LOCATE7,14:COLOR6:PRINT"♥♥♥♥ ";:COLOR7:PRINT"ｶﾌﾞ ";:COLOR6
32 PRINT"♥♥♥♥";SPC(5);:COLOR4:PRINT"by ";:COLOR6:PRINT"PALCOM":PRINT
33 COLOR7:PRINT"  ｺﾚｶﾗ ﾉ ";:COLOR3:PRINT"CARD ";:COLOR7
34 PRINT"ｹﾞｰﾑ ﾉ ﾋﾄﾂ ･････";:COLOR2:PRINT "ｶﾌﾞ";:COLOR7:PRINT"ﾃﾞｽ":PRINT
35 PRINT" FM-8 ｶﾞ ｱﾅﾀ ﾉ ｱｲﾃ ｦ ｼﾏｽ｡(";:COLOR2:PRINT"3";
36 COLOR7:PRINT"ﾆﾝ ﾏﾃﾞOK !!)"
37 PRINT" ﾊﾀｼﾃ ｱﾅﾀ ﾊ FM-8 ﾆ ｳﾁｶﾂｺﾄｶﾞﾃﾞｷﾙｶ ???":PRINT
38 PRINT SPC(6);:COLOR5:PRINT"GAME START ";:COLOR6:PRINT"･･･ ";
39 LOCATE 22,22:COLOR4:PRINT"Hit CR key !!":FOR I=1 TO 180:NEXT
```

```
40 IF INKEY$<>CHR$(13) THEN LOCATE 22,22:PRINT SPC(13):FOR I=1 TO 160:NEXT:GOTO
39
41 ' --  ｹﾞｰﾑ ﾉ ｾﾂﾒｲ  --
42 BEEP:COLOR4,0:CLS:LOCATE 9,24:PRINT"♠♥♦♣";:COLOR1:PRINT" ﾙｰﾙ ﾉ ｾﾂﾒｲ ";:COLOR4
43 PRINT"♣♦♥♠":PRINT:PRINT:COLOR7
44 PRINT"  4ﾏｲ ﾉ ｶｰﾄﾞ ｦ ﾅﾗﾍﾞﾏｽ｡ｽｷﾅ ｶｰﾄﾞﾆ ﾃﾝｽｳ ｦ":PRINT
45 PRINT"  ｶｹﾃ ｸﾀﾞｻｲ｡ｱﾅﾀ ﾉ ﾓﾁﾃﾝﾊ 3000ﾃﾝ ﾃﾞｽ｡":PRINT
46 PRINT"  9ﾆ ﾁｶｲｶｽﾞ ｦ ﾀﾞｼﾀﾎｳｶﾞ ｶﾁﾃﾞｽ｡":PRINT
47 PRINT"  ｶｯﾀ ﾄｷﾆﾊ ?ﾊﾞｲ ﾆ ﾅｯﾃ ｶｴｯﾃｷﾏｽ｡ﾄﾞﾁﾗｶﾉ":PRINT
48 PRINT"  ﾃﾝｽｳｶﾞ ﾅｸﾅﾚﾊﾞ ｹﾞｰﾑ ﾊ ｵﾜﾘﾃﾞｽ｡":PRINT:PRINT
49 PRINT SPC(8);:COLOR3:PRINT"*** ";:COLOR5:PRINT"ﾊﾞｲﾘﾂ";:COLOR3:PRINT" ***";
50 COLOR6:PRINT" (ﾂﾖｲ ｼﾞｭﾝ)":PRINT
51 PRINT"    1)";:COLOR7:PRINT"  ｽﾘｰｶｰﾄﾞ (";:COLOR2:PRINT"ｱﾗｼ";:COLOR7
52 PRINT") ....  ";:COLOR5:PRINT"3";:COLOR7:PRINT"ﾊﾞｲ":COLOR6:PRINT
53 PRINT"    2)";:COLOR7:PRINT"  9 & 1 (";:COLOR2:PRINT"ｸｯﾋﾟﾝ";:COLOR7
54 PRINT") ....  ";:COLOR5:PRINT"2";:COLOR7:PRINT"ﾊﾞｲ":COLOR6:PRINT
55 PRINT"    3)";:COLOR7:PRINT"  4 & 1 (";:COLOR2:PRINT"ｼｯﾋﾟﾝ";:COLOR7
56 PRINT") ....  ";:COLOR5:PRINT"2";:COLOR7:PRINT"ﾊﾞｲ":COLOR6:PRINT:PRINT
57 LOCATE 12,23:PRINT"Hit CR key !!":FOR I=1 TO 180:NEXT
58 IF INKEY$<>CHR$(13) THEN LOCATE 12,23:PRINT SPACE$(13):FOR I=1 TO 150:NEXT:GO
TO 57
59 ' --  ﾆﾝｽﾞｳ input  --
60 BEEP:COLOR 7:CLS
61 LOCATE 0,1:PRINT "ｺﾉ ｹﾞｰﾑ ｦ ﾅﾝﾆﾝﾃﾞｵｺﾅｲﾏｽｶ ? (1-3)          ";
62 LOCATE 32,1:INPUT " ",NINZU$
63 IF NINZU$="1" OR NINZU$="2" OR NINZU$="3" THEN NINZUU=VAL(NINZU$) ELSE 61
64 FOR II=0 TO 2:NAM$(II)="":NEXT
66 FOR II=1 TO NINZUU:Y=(II-1)*2+5
68 LOCATE 0,Y:PRINT"ﾅﾏｴｦ ｲﾚﾃｸﾀﾞｻｲ ";:COLOR6:PRINT"(PLAYER";II;")";:COLOR7
69 INPUT A$:NAM$(II-1)=LEFT$(A$+SPACE$(7),7):A$=""
72 NEXT
100
105 '      ****  GAME START  ****
110
115 GOSUB 1000
120 GOSUB 1500
125 GOSUB 2000
130 ' --  ｶｹﾃﾝ input  --
135 I1=0
140 I1=I1+1
145 IF I1>NINZUU THEN 300
150 IF TEN(I1-1)<=0 GOTO 140
155 LOCATE 44,17:PRINT"***  ";NAM$(I1-1);:BEEP
160 LOCATE 59,19:PRINT SPACE$(8);:LOCATE 59,20:PRINT SPACE$(8);
165 LOCATE 52,19:PRINT"No?          ":LOCATE 52,20:PRINT"ｶｹﾃﾝ?"
170 LOCATE 55,19:INPUT " ",W$
171 IF (W$="1") OR (W$="2") OR (W$="3") OR (W$="4") THEN W1=VAL(W$) ELSE 170
175 LOCATE 52,20:PRINT"ｶｹﾃﾝ?              "
180 LOCATE 57,20:INPUT " ",W$:IF W$="" THEN 180
185 W3=LEN(W$):FOR I=1 TO W3
190 IF (ASC(MID$(W$,I,1))<ASC("0"))OR(ASC(MID$(W$,I,1))>ASC("9")) THEN 175
195 NEXT I:W2=VAL(W$):IF (W2<0) OR (W2>TEN(I1-1)) THEN 175
196 LOCATE 66,(I1-1)*2+8:PRINT W1;:LOCATE 72,(I1-1)*2+8:PRINTUSING"#####";W2;
197 ZXX(I1-1,0)=W1:ZXX(I1-1,1)=W2
200 LOCATE 49,17:PRINT SPACE$(10);:LOCATE 59,19:PRINT SPACE$(10);
205 LOCATE 58,20:PRINT SPACE$(15);:GOTO 140
290 '
300 LOCATE 44,17:PRINT SPACE$(30);
305 LOCATE 44,19:PRINT SPACE$(30);
310 LOCATE 44,20:PRINT SPACE$(30);
315 LOCATE 50,19:PRINT"INPUT OK? (Y/N)";
320 W$=INKEY$:IF W$="" THEN 320
325 IF (W$="n") OR (W$="N") THEN LOCATE50,19:PRINT SPACE$(10):GOSUB 9010
330 '  --  ｶｰﾄﾞ ｦ ｸﾊﾞﾙ  --
335 FOR I1=1 TO 4:FOR J=0 TO 2
340 IF ZXX(J,0)=I1 THEN 400
345 NEXT J
350 CT=CT+1:CR=C1(CT):X=(I1-1)*8+5:Y=14:CARD(2,I1)=CR:GOSUB 5000
355 W0=CARD(1,I1) MOD 10:W1=CARD(2,I1) MOD 10:W2=W1+W0
360 IF W2=5 THEN IF (W0=1) OR (W1=1) THEN CARD(3,I1)=0:GOTO 450
365 W1=(CARD(1,I1)+CARD(2,I1)) MOD 10
370 IF W1>=7 THEN CARD(3,I1)=0:GOTO 450
375 CT=CT+1:CR=C1(CT):X=(I1-1)*8+5:Y=19:CARD(3,I1)=CR:GOSUB 5000
380 GOTO 450
400 LOCATE 45,17:PRINT"*******    <";I1;">  ";
```

```
401 IF INKEY$<>"" THEN 401
402 CT=CT+1:CR=C1(CT):X=(I1-1)*8+5:Y=14:CARD(2,I1)=CR:GOSUB 3000
404 LOCATE 50,19:PRINT"ｺﾏｲﾒﾉ ｶｰﾄﾞｦ ﾋｷﾏｽｶ (Y/N) ?";
406 A$=INKEY$:IF (A$="N") OR (A$="n") THEN CARD(3,I1)=0:GOTO 430
408 IF NOT((A$="Y") OR (A$="y") OR (A$=CHR$(13))) THEN 406
410 LOCATE 50,19:PRINT SPACE$(30)
412 CT=CT+1:CR=C1(CT):X=(I1-1)*8+5:Y=19:CARD(3,I1)=CR:GOSUB 3000
430 FOR TI=1 TO 1500:NEXT
435 LOCATE 45,17:PRINT SPACE$(30);:LOCATE 50,19:PRINT SPACE$(30);
440 Y=14:GOSUB 4000
450 NEXT I1
500 ' ---  ﾄﾞｳﾓﾄ ﾆ ｶｰﾄﾞｦ ｾｯﾄ  ---
505 GOSUB 6000
510 ' ---  card open  ---
515 FOR I1=1 TO 4
520 IF AAX(I1)=1 THEN 600
525 FOR J1=0 TO 2
530 IF ZXX(J1,0)=I1 THEN 545
535 NEXT J1:GOTO 560
545 CR=CARD(2,I1):X=(I1-1)*8+5:Y=14:GOSUB 3000
550 FOR TI=0 TO 500:NEXT
555 GOTO 600
560 FOR J2=2 TO 3
565 CR=CARD(J2,I1):IF CR<>0 THEN X=(I1-1)*8+5:Y=(J2-2)*5+14:GOSUB 3000
570 FOR TI=0 TO 500:NEXT
575 NEXT J2
600 NEXT I1
605 IF SC=0 THEN I1=0 ELSE I1=2
610 I1=I1+1
615 IF I1>3 THEN 650 ELSE CR=CB(I1)
620 IF CR<>0 THEN X=(I1-1)*8+5:Y=2:GOSUB 3000:FOR TI=0 TO 500:NEXT:GOTO 610
625 GOTO 610
650 ' ---  ﾃﾝｽｳ ｹｲｻﾝ  ---
655 W1=CB(1) MOD 10:W2=CB(2) MOD 10:W5=(W1+W2) MOD 10
660 IF (CB(3)=0) THEN SC=2:IF W5=0 THEN S9=2:GOTO 690 ELSE S9=1:GOTO 690
665 SC=3:W3=CB(3) MOD 10
670 IF (W1=W2) AND (W2=W3) THEN S9=3:GOTO 690
675 W5=(W5+W3) MOD 10
680 IF W5=0 THEN S9=0 ELSE S9=1
690 FOR J8=1 TO 4
700 IF AAX(J8)=1 THEN 890
710 FOR J7=0 TO 2
720 IF ZXX(J7,0)<>J8 THEN 880
730 W1=CARD(1,J8) MOD 10:W2=CARD(2,J8) MOD 10
740 IF CARD(3,J8)<>0 THEN 770
750 IF (W1+W2=5) AND ((W1=1) OR (W2=1)) THEN S8=2:GOTO 810
755 WW=(W1+W2) MOD 10
760 IF WW=0 THEN S8=0:GOTO 810 ELSE S8=1:GOTO 810
770 W3=CARD(3,J8) MOD 10
780 IF (W1=W2) AND (W1=W3) THEN S8=3:GOTO 810
790 WW=(W1+W2+W3) MOD 10
800 IF WW=0 THEN S8=0 ELSE S8=1
810 IF (S9=0) OR (S8=0) THEN SS=0 :GOTO 860
815 IF (S9=2) AND (S8=2) THEN SS=-2:GOTO 860
820 IF (S9=3) AND (S8=3) THEN SS=0 :GOTO 860
830 IF S8>S9 THEN SS=S8            :GOTO 860
840 IF S9>S8 THEN SS=-1*S9         :GOTO 860
850 IF WW=W5 THEN SS=0 :GOTO 860
855 IF WW>W5 THEN SS=1 ELSE SS=-1
860 CCX(J7,0)=ZXX(J7,1)*SS:CCX(J7,1)=SC
880 NEXT J7
890 NEXT J8
891 FOR J7=15 TO 23
892  LOCATE 42,J7:PRINT SPACE$(36);
893 NEXT:W1=0
894 FOR J7=0 TO 2
895  IF TEN(J7)=<0 THEN901
896  W1=W1+1:LOCATE 45,14+2*W1:PRINT NAM$(J7);" ｻﾝ  ";
897  IF CCX(J7,0)<0 THEN COLOR 2,0 ELSE COLOR 7,0
898  PRINT USING "+######    ";CCX(J7,0);:COLOR 7,0
899  PRINT CCX(J7,1);"ﾏｲﾒﾉ ｼｮｳﾌﾞ";
901 NEXT J7
902 LOCATE 48,22:PRINT"Hit CR key  !!";:FOR TI=0 TO 180:NEXT
903 IF INKEY$<>CHR$(13) THEN LOCATE48,22: PRINTSPACE$(14): FOR TI=0 TO 150:NEXT:
    GOTO902
910 '
```

```
911 FOR I1=0 TO 2
912  IF TEN(I1)<=0 THEN 915
913  TEN(I1)=TEN(I1)+CCX(I1,0)
914  LOCATE 57,I1*2+8:PRINT USING "  ######";TEN(I1);
915 NEXT I1
916 TEN(3)=TEN(3)-CCX(0,0)-CCX(1,0)-CCX(2,0):FOR I1=0 TO 2:CCX(I1,0)=0:NEXT
917 LOCATE 57,6:PRINT USING "  ######";TEN(3);
920 IF TEN(3) <= 0 THEN 8000
925 IF (TEN(0)<=0) AND (TEN(1)<=0) AND (TEN(2)<=0) THEN 8000
930 '
960 FOR I1=15 TO 23:LOCATE 40,I1:PRINT SPACE$(40);:NEXT
965 FOR I1=5 TO 13:LOCATE 66,I1:PRINT SPACE$(3);
970 LOCATE 71,I1:PRINT SPACE$(6):NEXT
975 FOR I1=2 TO 24:LOCATE 3,I1:PRINT SPACE$(37);:NEXT I1
976 LINE(0,55)-(320,55),PSET,0 ──── 7, 8は不要 (REM文に)
980 GOTO 120
1000 '
1010 '   *****  ｶﾞﾒﾝ 1  *****
1020 '
1030 WIDTH 80,25:CLS
1040 LOCATE 4,0:COLOR2:PRINT"♥♥♥♥♥♥♥♥♥♥♥    ";:COLOR6::PRINT"ｶﾌﾞ";
1050 COLOR2:PRINT"    ♥♥♥♥♥♥♥♥♥♥♥":COLOR5:LOCATE 44,1
1060 PRINT"     <<< ";:COLOR4:PRINT"  ﾄｸﾃﾝ ﾋｮｳ";:COLOR5:PRINT"    >>>":COLOR6
1070 LOCATE44,2:PRINT"┌─────────┬─────────┬────┬─────┐"
1080 LOCATE44,3:PRINT"│  ﾅ ﾏ ｴ   │ ﾃﾝｽｳｻﾞﾝ │ No │ ｶｹﾃﾝ │"
1090 LOCATE44,4:PRINT"├─────────┼─────────┼────┼─────┤"
1100 FOR P=5 TO 13
1110 LOCATE44,P:PRINT"│         │         │    │     │"
1120 NEXT
1130 LOCATE44,14:PRINT"└─────────┴─────────┴────┴─────┘"
1140 LOCATE 46,6:PRINT"ﾄﾞ ｳ ﾓ ﾄ"
1150 TEN(3)=10000*NINZUU:LOCATE 58,6:PRINT TEN(3)
1160 FOR I1=0 TO 2
1170 IF NAM$(I1)="" THEN TEN(I1)=0:GOTO 1190
1180 TEN(I1)=3000:LOCATE46,I1*2+8:PRINTNAM$(I1);:LOCATE59,I1*2+8:PRINT TEN(I1);
1190 NEXT
1200 RETURN
1500 '
1510 '    *****  ｶｰﾄﾞ ｦ ｸﾊﾞﾙ  *****
1520 '
1530 LOCATE 45,19:COLOR1:PRINT"ﾀﾀﾞｲﾏ ｶｰﾄﾞｦ ｷｯﾃｲﾏｽ !!";:COLOR7
1540 FOR I=1 TO 40:SW(I)=0:C1(I)=0:NEXT
1560 FOR J=1 TO 40
1570 I=INT(RND(1)*40)+1:IF SW(I)=1 THEN 1570
1580 C1(J)=I:SW(I)=1:NEXT
1590 LOCATE45,19:PRINT SPACE$(30)
1600 RETURN
2000 '
2010 '    *****  ｶｰﾄﾞ ｦ ｵｸ ( 1 ｶｲﾒ )  *****
2020 '
2030 LOCATE6,8:PRINT"[1]      [2]      [3]      [4]"
2040 FOR I=0 TO 3:M=((I-2)*5+19)*8:LINE(28,M-1)-(284,M-1),PSET,2:NEXT
2050 FOR I=28 TO 284 STEP 64:LINE(I,8*19-80)-(I,8*19+39),PSET,2:NEXT
2060 FOR I=0 TO 3:FOR J=0 TO 4:CARD(I,J)=0:NEXT J,I
2070 FOR I=0 TO 3:CB(I)=0:NEXT
2080 FOR I=0 TO 4:AAX(I)=0:BBX(I)=0:NEXT:SC=0:CT=0
2090 FOR I=0 TO 2:FOR J=0 TO 1:ZXX(I,J)=0:NEXT J,I
2100 CT=CT+1
2110 IF CT>4 THEN 2130
2120 CR=C1(CT):X=(CT-1)*8+5:Y=9:GOSUB 3000:CARD(1,CT)=CR:GOTO 2100
2130 X=5:Y=2:GOSUB 4000:CB(1)=C1(CT)
2140 RETURN
3000 '
3010 '   *****  card pattern  *****
3020 '
3030 C=INT((CR-1)/10)+1:ON C GOTO 3040,3050,3060,3070
3040 M$="♠":IR=0:GOTO 3080
3050 M$="♥":IR=1:GOTO 3080
3060 M$="♦":IR=1:GOTO 3080
3070 M$="♣":IR=0:GOTO 3080
3080 COLOR 7
3100 C=CR-C*10+10:ON C GOTO 3150,3200,3250,3300,3350,3400,3450,3500,3550,3600
3150 COLOR 15
3155 LOCATE X,Y  :PRINT"    A";
3157 LOCATE X,Y+1:PRINT"     ";
```

```
3160 LOCATE X,Y+2:PRINT"  ";M$;"  ";
3165 LOCATE X,Y+3:PRINT"     ";
3170 LOCATE X,Y+4:PRINT"A    ":GOTO 3700
3200 COLOR 15
3210 LOCATE X,Y  :PRINT"    2";
3220 LOCATE X,Y+1:PRINT"  ";M$;"  ";
3225 LOCATE X,Y+2:PRINT"     ";
3230 LOCATE X,Y+3:PRINT"  ";M$;"  ";
3240 LOCATE X,Y+4:PRINT"2    ":GOTO 3700
3250 COLOR 15
3260 LOCATE X,Y  :PRINT"    3";
3270 LOCATE X,Y+1:PRINT"  ";M$;"  ";
3280 LOCATE X,Y+2:PRINT"  ";M$;"  ";
3290 LOCATE X,Y+3:PRINT"  ";M$;"  ";
3295 LOCATE X,Y+4:PRINT"3    ":GOTO 3700
3300 COLOR 15
3310 LOCATE X,Y  :PRINT"    4";
3320 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3325 LOCATE X,Y+2:PRINT"     ";
3330 LOCATE X,Y+3:PRINT" ";M$;" ";M$;" ";
3340 LOCATE X,Y+4:PRINT"4    ":GOTO 3700
3350 COLOR 15
3360 LOCATE X,Y  :PRINT"    5";
3370 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3375 LOCATE X,Y+2:PRINT"  ";M$;"  ";
3380 LOCATE X,Y+3:PRINT" ";M$;" ";M$;" ";
3390 LOCATE X,Y+4:PRINT"5    ":GOTO 3700
3400 COLOR 15
3410 LOCATE X,Y  :PRINT"    6";
3420 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3425 LOCATE X,Y+2:PRINT" ";M$;" ";M$;" ";
3430 LOCATE X,Y+3:PRINT" ";M$;" ";M$;" ";
3440 LOCATE X,Y+4:PRINT"6    ":GOTO 3700
3450 COLOR 15
3460 LOCATE X,Y  :PRINT"    7";
3470 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3475 LOCATE X,Y+2:PRINT" ";M$;" ";M$;" ";
3480 LOCATE X,Y+3:PRINT"  ";M$;"  ";
3490 LOCATE X,Y+4:PRINT"7";M$;" ";M$;" ":GOTO 3700
3500 COLOR 15
3510 LOCATE X,Y  :PRINT"    8";
3520 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3525 LOCATE X,Y+2:PRINT" ";M$;" ";M$;" ";
3530 LOCATE X,Y+3:PRINT" ";M$;" ";M$;" ";
3540 LOCATE X,Y+4:PRINT"8";M$;" ";M$;" ":GOTO 3700
3550 COLOR 15
3560 LOCATE X,Y  :PRINT"    9";
3570 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3575 LOCATE X,Y+2:PRINT" ";M$;M$;M$;" ";
3580 LOCATE X,Y+3:PRINT" ";M$;" ";M$;" ";
3590 LOCATE X,Y+4:PRINT"9";M$;" ";M$;" ":GOTO 3700
3600 COLOR 15
3610 LOCATE X,Y  :PRINT"   10";
3620 LOCATE X,Y+1:PRINT" ";M$;" ";M$;" ";
3625 LOCATE X,Y+2:PRINT" ";M$;M$;M$;" ";
3630 LOCATE X,Y+3:PRINT" ";M$;M$;M$;" ";
3640 LOCATE X,Y+4:PRINT" ";M$;" ";M$;" ":GOTO 3700
3700 FOR I9=0 TO 80:BEEP 1:BEEP 0:BEEP 0:NEXT
3710 COLOR 2,0
3720 IF IR=0 THEN 3740                     ——⑦, ⑧は不要
3730 LINE@ ((X+1)*8,(Y+1)*8)-((X+5)*8-1,(Y+5)*8-1),OR,2,BF
3740 LINE@ (X*8,(Y+5)*8-1)-((X+5)*8,(Y+5)*8-1),PSET,2   ——⑦, ⑧は不要
3750 COLOR 7,0
3760 RETURN
4000 '
4010 '  *****  card ウラ  *****
4020 '
4030 COLOR 5
4040 Q$=STRING$(5,CHR$(&HFE))
4050 LOCATE X,Y  :PRINT Q$;
4060 LOCATE X,Y+1:PRINT Q$;
4070 LOCATE X,Y+2:PRINT Q$;
4080 LOCATE X,Y+3:PRINT Q$;
4090 LOCATE X,Y+4:PRINT Q$;
4092 COLOR 2,0
4094 LINE@ (X*8,(Y+5)*8-1)-((X+5)*8-1,(Y+5)*8-1),PSET,2   ——⑦, ⑧は不要
4100 COLOR 7
```

```
4110 FOR TI=0 TO 80:BEEP1:BEEP 0:BEEP 0:NEXT TI
4120 RETURN
5000 '
5010 '    *****  card ｼﾛ  *****
5020 '
5030 COLOR 7
5040 LOCATE X,Y  :PRINT"█████";
5050 LOCATE X,Y+1:PRINT"█████";
5060 LOCATE X,Y+2:PRINT"█████";
5070 LOCATE X,Y+3:PRINT"█████";
5080 LOCATE X,Y+4:PRINT"█████";
5092 COLOR 2,0
5094 LINE@ (X*8,(Y+5)*8-1)-((X+5)*8-1,(Y+5)*8-1),PSET,2
5096 COLOR 7
5098 FOR TI=0 TO 20:BEEP 1:BEEP 0:NEXT
5100 RETURN
6000 '
6010 '   *****  ﾄﾞｳﾓﾄ  *****
6020 '
6030 CT=CT+1:X=13:Y=2:CB(2)=C1(CT):GOSUB 4000
6040 W=CB(1) MOD 10:W1=CB(2) MOD 10:IF (W+W1)=10 THEN 6060
6050 GOTO 6070
6060 IF (W1=1) OR (W1=9) THEN CB(3)=0:GOTO 6990
6070 W2=W+W1:W5=W2 MOD 10
6080 IF W5 >= 7 THEN CB(3)=0:GOTO 6990
6090 IF W5 <= 4 THEN 6800
6100 FOR J8=1 TO 4
6110 IF AAX(J8)=1 THEN 6200
6120 FOR J7=0 TO 2
6130 IF ZXX(J7,0)=J8 THEN 6160
6140 NEXT J7
6150 GOTO 6200
6160 IF CARD(3,J8)=0 THEN 6200
6170 W6=CARD(3,J8) MOD 10
6180 IF (W6=9) OR (W6=0) OR (W6=1) THEN BBX(J8)=1:GOTO 6200
6190 W=INT(10*RND):IF (5<=W) AND (W<=9) THEN BBX(J8)=1
6200 NEXT J8
6210 '
6220 FOR J7=1 TO 4
6230 IF BBX(J7)<>0 THEN 6250
6240 NEXT J7:GOTO 6800
6250 FOR J7=16 TO 19:LOCATE 40,J7:PRINT SPACE$(40);:NEXT J7
6260 LOCATE 45,17:PRINT"***  2ﾏｲﾒﾃﾞ ｼｮｳﾌﾞ ｼﾏｽ   !!";:W4=0:SC=1
6270 FOR J7=1 TO 4
6280 IF BBX(J7)=0 THEN 6283
6281 W4=W4+1
6282 LOCATE (W4-1)*6+50,19:PRINT"[";J7;"]";
6283 NEXT J7
6285 X=5:Y=2:CR=CB(1):GOSUB 3000:X=13:CR=CB(2):GOSUB 3000
6287 FOR J7=1 TO 4
6289 IF BBX(J7)=0 THEN 6600
6290 AAX(J7)=1
6300 X=(J7-1)*8+5:Y=14:CR=CARD(2,J7):GOSUB 3000
6340 W1=CARD(1,J7) MOD 10:W2=CARD(2,J7) MOD 10:W3=CARD(3,J7) MOD 10
6350 IF (W1=W2) AND (W1=W3) THEN SS=3:GOTO 6400
6360 WW=(W1+W2+W3) MOD 10
6370 IF (WW=0) OR (WW=W5) THEN SS=0:GOTO 6400
6380 IF WW<W5 THEN SS=-1 ELSE SS=1
6400 FOR J5=0 TO 2
6410 IF ZXX(J5,0)<>J7 THEN 6500
6420 CCX(J5,0)=ZXX(J5,1)*SS:CCX(J5,1)=2
6500 NEXT J5
6600 NEXT J7
6620 FOR TI=0 TO 1000:NEXT TI
6630 LOCATE 45,17:PRINT SPACE$(34);
6640 LOCATE 45,19:PRINT SPACE$(34);
6650 '
6800 CT=CT+1:X=21:Y=2:CB(3)=C1(CT):GOSUB 4000
6990 RETURN
8000 '
8010 '   *****  END  *****
8020 '
8021 FOR I1=15 TO 23:LOCATE 40,I1:PRINT SPACE$(40);:NEXT
8030 LOCATE 50,22:PRINT"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ (Y/N) ?";
8040 A$=INKEY$:IF A$="" THEN 8040
8050 IF (A$="Y") OR (A$="y") OR (A$=CHR$(13)) THEN RUN
```

```
8060 '
8070 WIDTH 80,25:CLS
8080 LOCATE 25,11:PRINT"╭──────────────────────────╮"
8090 LOCATE 25,12:PRINT"│ ｿﾉﾀ ";:COLOR 2:PRINT"GAME SOFT ";:COLOR 7
8100 PRINT"ﾉ ｺﾞﾖｳﾒｲﾊ          │"
8110 LOCATE 25,13:PRINT"│ ";:COLOR 2:PRINT"SOFT ";:COLOR 7:PRINT"ｶｲﾊﾂ ﾉ";
8120 COLOR 6:PRINT" PALCOM ";:COLOR 7:PRINT"ﾍ ﾄﾞｳｿﾞ  │"
8130 LOCATE 25,14:PRINT"╰──────────────────────────╯":BEEP:END
9010 '
9020 LOCATE 66, 8:PRINT SPACE$(3);:LOCATE 71, 8:PRINT SPACE$(6)
9030 LOCATE 66,10:PRINT SPACE$(3);:LOCATE 71,10:PRINT SPACE$(6)
9040 LOCATE 66,12:PRINT SPACE$(3);:LOCATE 71,12:PRINT SPACE$(6)
9050 RETURN 130
10000 '
10010 FOR I=0 TO 64
10020  READ A$,B$
10030  IF VAL("&H"+A$)<&H80 THEN 10050
10040  OICH%(I)=-(&HFFFF-(VAL("&H"+A$+B$)))-1:GOTO 10060
10050  OICH%(I)=VAL("&H"+A$+B$)
10060 NEXT I
10070 PUT@ (272,0)-(335,15),OICH%,PSET,2
10500 COLOR6
10510 LOCATE 4,2:PRINT"    ╭─╮  ╭─╮       ╭─╮    "
10520 LOCATE 4,3:PRINT"    │";:COLOR 3:PRINT"■";:COLOR 6:PRINT"│  │";:COLOR 3
10530 PRINT"■";:COLOR 6:PRINT"╲        │";:COLOR 3:PRINT"■";
10540 COLOR 6:PRINT"╲"
10550 LOCATE 4,4:PRINT"╭─╯";:COLOR 3:PRINT"■";:COLOR 6:PRINT"╰──┴╮";
10560 COLOR 3:PRINT"■";:COLOR 6:PRINT"╲        ╮";
10570 COLOR 3:PRINT"■";:COLOR 6:PRINT"├─┬─╮"
10580 LOCATE 4,5:PRINT"│";:COLOR 3:PRINT"█████████";:COLOR 6:PRINT"├╮";
10590 COLOR 3:PRINT"■";:COLOR 6:PRINT"│        ╭┴┬┤";:COLOR 3:PRINT"■";
10600 COLOR 6:PRINT"│";:COLOR 3:PRINT"■";:COLOR 6:PRINT"│"
10610 LOCATE 4,6:PRINT"╰──╮";:COLOR 3:PRINT"■";:COLOR 6:PRINT"╭─╮";
10620 COLOR 3:PRINT"■";:COLOR 6:PRINT"│╰─╯       │";
10630 COLOR 3:PRINT"■";:COLOR 6:PRINT"│╰─┴─╯"
10640 LOCATE 4,7:PRINT"   │";:COLOR 3:PRINT"■";:COLOR 6:PRINT"│ │";
10650 COLOR 3:PRINT"■";:COLOR 6:PRINT"│        ╭─╮ │";
10660 COLOR 3:PRINT"■";:COLOR 6:PRINT"│ ╭─╮"
10670 LOCATE 4,8:PRINT"  ╱";:COLOR 3:PRINT"■";:COLOR 6:PRINT"│ │";
10680 COLOR 3:PRINT"■";:COLOR 6:PRINT"│        ╱";:COLOR 3:PRINT"■";
10690 COLOR 6:PRINT"│ │";:COLOR 3:PRINT"■";:COLOR 6:PRINT"╲│";
10700 COLOR 3:PRINT"■";:COLOR 6:PRINT" ╲"
10710 LOCATE 4,9:PRINT"  │";:COLOR 3:PRINT"■";:COLOR 6:PRINT"╱ │";
10720 COLOR 3:PRINT"■";:COLOR 6:PRINT"│        │";:COLOR 3:PRINT"■";
10730 COLOR 6:PRINT"╱ ╲";:COLOR 3:PRINT"■";:COLOR 6:PRINT"│╲";
10740 COLOR 3:PRINT"■";:COLOR 6:PRINT"│"
10750 LOCATE 4,10:PRINT" ╱";:COLOR 3:PRINT"■";:COLOR 6:PRINT"│ ╱";
10760 COLOR 3:PRINT"■";:COLOR 6:PRINT"│ ╰─╯╭─╯";
10770 COLOR 3:PRINT"■";:COLOR 6:PRINT"│ ╰─╯"
10780 LOCATE 4,11:PRINT" │";:COLOR 3:PRINT"■";:COLOR 6:PRINT"╱ │";
10790 COLOR 3:PRINT"██";:COLOR 6:PRINT"│        │";:COLOR 3:PRINT"██";
10800 COLOR 6:PRINT"╱"
10810 LOCATE 4,12:PRINT"╰─╯  ╰──╯        ╰──╯":RETURN
11000 DATA 03,00,00,00,01,80,00,00,03,00,00,00,01,80,00,00
11010 DATA 03,0C,00,38,03,00,00,00,03,0C,00,38,03,00,00,00
11020 DATA 3F,F3,30,0C,3F,F0,03,00,3F,F3,30,0C,3F,F0,03,00
11030 DATA 03,00,30,0C,0C,00,03,F0,03,00,30,0C,0C,00,03,F0
11040 DATA 7F,F8,30,0C,1F,F8,03,00,7F,F8,30,0C,1F,F8,03,00
11050 DATA C3,0C,3C,0C,00,0C,1F,C0,C3,0C,3C,0C,00,0C,1F,C0
11060 DATA C3,0C,38,00,00,0C,33,30,C3,0C,38,00,00,0C,33,30
11070 DATA 7E,38,00,00,0F,F8,1F,00,7E,38,00,00,0F,F8,1F,00,00,00,00
```

BASICチェックサム（44ページ参照）

□で囲まれた部分はFM-7，8用

```
LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
   1 00 00 00 00 00 00 00 00 :00
   9 00 00 76 A8 16 DD 21 71 :A3
  31 96 36 3A 39 50 D5 9D 0D :0E
  39 92 0A 00 EB B1 EB 49 D8 :44
  47 93 0A 33 42 84 AF B8 AE :AB
  55 B8 75 A5 B3 00 28 D4 48 :C9
  63 17 8E 40 B6 34 3F 00 00 :0E
 110 00 41 46 42 00 E7 8D F2 :2F
 150 16 2E E0 74 49 C4 E2 8C :13
 185 77 59 DB 2C E4 37 04 00 :F6
--------------------------------
     17 15 C9 59 FC 95 06 CA /AF
```

```
                              00
LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
 300 FA FC F4 85 A1 6D 00 53 :D0
 340 0A 89 EC 04 62 07 B5 F2 :93
 380 D2 F1 71 EA D9 3C 3D BE :2E
 412 F0 D4 2E 79 B9 00 46 00 :6A
 515 A6 67 A4 45 C8 81 A2 CF :B0
 560 A8 7E A2 BB B9 BA 8D 8F :12
 620 E2 D0 00 27 BD 51 DF 9C :62
 680 D3 AE 7A AA 92 DC 52 17 :7C
 755 B9 0B 54 D7 6E F3 16 7B :E1
 820 4E 92 1B 59 B4 6C C0 C1 :F5
--------------------------------
     D0 4A AE ED 87 77 6E 50 /71

                 00
LINE +0 +1 +2 +3|+4 +5 +6 +7 SUM
 891 13 17 6E AA|BA AB 7E 3E :63
 899 13 C0 5B 3E|00 A3 B8 D1 :98
 914 10 B9 D0 C6|9A 55 00 C8 :16
 965 F4 1D E4[AC]CC 00 00 00 :[6D]─C1
1030 97 3A 97 91 0A 6E FA B0 :1B
1110 F1 3F 3E 97 9F A3 23 B9 :23
1190 3F E0 00 00 00 A6 BE A6 :29
1570 89 FA C2 E0 00 00 00 A3 :C8
2040 FD DC 2B B2 A0 17 C7 8D :C1
2120 2C 24 E0 00 00 00 E5 16 :2B
--------------------------------
     A3 00 1F[14]69 71 BD 2C /[99]─ED
              68
LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
3050 18 19 19 B6 55 E5 58 93 :25
3160 5B 95 B9 E5 49 5A 94 5C :21
3240 AA E5 4A 5A 5B 5C AB E5 :7A
3310 4B 21 94 23 AC E5 4C 21 :21
3375 5B 23 AD E5 4D 21 22 23 :C3
3440 AE E5 4E 21 22 5C 3D E5 :A2
```

```
                     D0
3510 4F 21 22 23 3E| E5 50 21 :49
3575 73 23 3F E5 58| 21 73 74 :1A   ─B0
3640 26 ED 0D 94[00][0E]12 E0 :[B4]─26
4000 00 00 00 B4 AA 0C 68 69 :3B
--------------------------------
     59 ED 19 6E[54][1D]7F DB /[98]─0A
       0E      0E 24  BF
LINE +0|+1 +2 +3|+4 +5 +6 +7 SUM
4080 6A|6B 0D[6C]B6 A5 E0 00 :[89]─2B
5010 00|00 B6 3A 96 97 98 99 :4E
5092 0D[6C]B6 7C E0 00 00 00 :[8B]─2D
6030 55 E1 06 B6 9D BD C5 AE :BF
6110 A1 AA 82 C0 01 3F 57 43 :67
6190 D7 C1 00 AD E3 01 5D FC :82
6270 AD AC AF 29 C0 20 AD A5 :63
6290 1A 8F 67 23 6E E3 B2 A8 :DE
6410 BA 02 BE C0 6B FF 01 00 :A5
6800 55 E0 00 00 00 C8 FE AC :A7
--------------------------------
     1A[40]D5[51]46 03 4F 7F /[97]─DB
        E2    F3
LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
8050 56 00 97 F9 41 FB 7F FF :A0
8130 67 00 3A 8C 90 74 00 AA :DB
10020 13 F8 2C E2 88 66 B5 DB :97
10520 01 C6 37 0E BE DC 2C A6 :78
10600 0E E9 4A 7E F9 46 75 7F :F2
10680 52 3A 2C 80 AD 80 87 2D :19
10760 D9 79 89 A2 B7 31 C6 F6 :21
11020 A2 38 C0 CC 4E 36 00 00 :EA
    0 00 00 00 00 00 00 00 00 :00
    0 00 00 00 00 00 00 00 00 :00
--------------------------------
     AC 92 F3 E1 C2 DE 22 CC /A0
```

バックナンバー情報

◇Vol.1〜Vol.3　品切れ

◇Vol.4〜Vol.7, FILES Oh／FM (Vol.1〜Vol.3 合本) 在庫あり

バックナンバーご希望の方は，最寄りの書店に申しこむか，下記の常備店をご利用ください。またそれぞれの内容については，Vol. 7「INDEX」をご覧ください。

Oh／シリーズ バックナンバーフェアのお知らせ

開催期間　3月17日(土)〜4月1日(日)

開催会場　長野県飯田市平安堂書店飯田店 2Fパソコン売場

(飯田市中央通り4−2☎0265-24-4545)

出展雑誌　Oh／FM　Oh／PC　Oh／MZ

バックナンバー常備店

神保町	三省堂神田本店5F	☎ 03-233-3312
神保町	書泉ブックマートB1	☎ 03-294-0011
八重洲	八重洲ブックセンター3F	☎ 03-281-1811
新宿	紀伊国屋書店	☎ 03-354-0131
新宿	未来堂書店	☎ 03-200-9185
渋谷	東急ハンズ7F寿楽洞	☎ 03-464-4604
池袋	西武百貨店9Fマイコン売場	☎ 03-981-0111
調布	真光書店	☎0424-87-2222
町田	東急ハンズ寿楽洞	☎0427-28-2782
横浜	有隣堂ルミネ店	☎045-453-0811
藤沢	有隣堂藤沢店	☎0466-26-1411
津田沼	芳林堂書店パルコ5F	☎0474-78-3737
柏	新星堂書店カルチェ5	☎0471-64-8551
船橋	西武ブックセンター10F	☎0474-25-0111
川口	岩渕書店	☎0482-52-2190
川越	黒田書店	☎0492-25-3138
水戸	川又書店駅前店	☎0292-31-0102
豊橋	精文館書店	☎0532-54-2345
飯田	平安堂飯田店	☎0265-24-4545
京都・中京区	オーム社書店	☎075-221-0280
大阪・都島区	駸々堂京橋店	☎ 06-353-2413

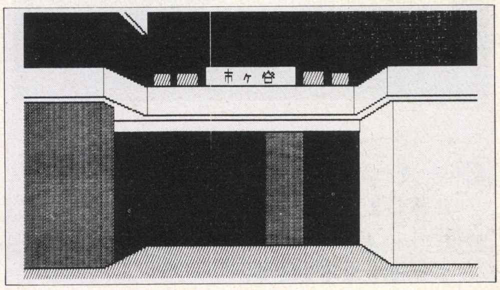
君もアドベンチャゲームを作ってみよう

for FM-7,8

アドベンチャゲーム開発ツール

米村　英明

マシン語の高速ゲームは作れないが，BASICのアドベンチャゲームなら作れそうだ，ストーリアイデアは，いっぱいある，という読者も多いことでしょう。しかし，いざ作ろうとすると，絵のデータをどうやってつくるか，キーワードをどう隠すかなどいろいろな問題が出てきますが，今回はそれらを解決してくれる開発ツールを紹介しましょう。



SHIFT＋Hでとったハードコピー

はじめに

序文でもちょっと触れましたが，アドベンチャゲームはBASICに多少慣れた人ならば何の苦労もなくメインとなるプログラムは組めてしまうでしょう。しかし，文字出力だけのテキストタイプのアドベンチャゲームはよほど内容を工夫していないかぎりつまらないものになってしまいます。やはり，アドベンチャゲームを作るならば絵を表示させたりしたいものです。逆にストーリはたいしたことはなくても，絵の表示などがすばらしければ，それだけでなかなかのものに見えてくるから不思議なものです。しかし，絵のデータというのはくせもので１枚で何十Ｋバイトものメモリを食ってしまったりします。また，せっかく凝った絵を描いても表示するのに時間がかかってしまってはゲームも退屈なものになってしまいます。

もう１つの問題はゲームの進行を操作するキーワードにあります。いかにおもしろいゲームであっても，キーを押して，リストをとってみたとき，キーワードや先の展開が簡単にばれてしまうようではどうしようもありません。市販のプログラムで多く見られるような，キーワードの文字変数への読み込みルーチンとゲームのメインプログラムをわけて実行するような方法だと，いちいち「あの変数にはあのメッセージが入っていて……この変数にはこのキーワードが入っていて……」ということを覚えておかなくてはならず，開発やデバッグのときパニックに陥ります。

そこで，高速，省メモリ型のマシン語プログラムを生成するピクチャエディタと簡単なキーワード隠しのユーティリティを製作しました。

アドベンチャゲーム・ユーティリティ

まず，**リスト１**を打ち込み，SAVE“PEDIT”でディスクまたはカセットにセーブしておきます。次に**リスト２**，**リスト３**のマシン語ルーチンを打ち込み，SAVEM“PD.SUB”，&H6000，&H6FD3,&H6E00でセーブします。なお，&H6100～&H65A1まではOh! FM Vol.6のタイルペイントルーチンをそのまま使っていますので，すでにこれを打ち込んである人はあらためて打ちなおす必要はありません。LOADM“TPAINT”で６号のプログラムをロードした後，&H6000～&H60F2までと&H6E00～&H6FD3までを打ち込み，前記のようにセーブしてください。なお，このユーティリティプログラムのメモリマップは**図１**のようになります。

ピクチャエディタ“PED IT”

このプログラムは，アドベンチャゲームには欠かせないグラフィック画面を作成するためのものです。省メモリを第一に考えて作ったプログラムで，普通のアドベンチャゲームに使われるくらいの絵なら１～３Ｋバイト，よほど，こまごま描き込んだものでも４～６Ｋバイト程度のマシン語サブルーチンの形に作成した絵を変換します。なお生成されるマシン語ルーチン自体はポジションインディペンデントで，どこのアドレスに置いても動きます。ただし，絵を表示するには$6E00からのランタイムルーチン“RUN. T”が必要で，また，中間色を使っている場合には“TPAINT”も必要になります。一度ランタイムルーチンなどをメモリ上にロードしておけば，“PEDIT”で作成したマシン語ルーチンはLOADM“ファイル名”,,Rで絵を描いてくれます。なお，このプログラムを使用するときは必ずcapキーを点灯させ，大文字にセットしてください。

1. ピクチャエディタのコマンド

❶ カーソル操作コマンド類

←→↑↓……カーソルを上下左右に動かします。

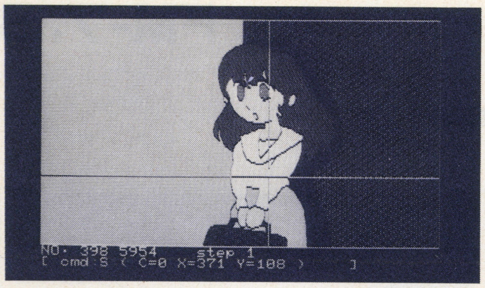
SHIFT＋1……カーソルの移動を１ドット単位にします。

SHIFT＋2……カーソルの移動を２ドット単位にします。

SHIFT＋4……カーソルの移動を４ドット単位にします。

SHIFT＋8……カーソルの移動を８ドット単位にします。

0～7……カーソルの色を変えます。



[A]……カーソルをx，yで指定した座標に移します。

[SHIFT]＋[A]……カーソルの現在のx，y座標を表示します。

❷ 実行コマンド類

[S]……PSETに相当します。カーソルのある所にカーソルの色で点を打ちます。また，この点がCONNECTで線を引くときの始点となります。

[C]……CONNECTに相当します。PSETで点を打った所，またはCONNECTで線を引いた終点から現在カーソルの指している所までカーソルの色で線を引きます。

[L]……PAINT，TPAINTを実行するときの境界色を指定します。4色まで指定できますが，4色も使わないといった場合でも，同じ色を2度打ち込むなどして，きっちり4色分入力してください。

[T]……TPAINTのときのタイルパターンを指定します。16進2桁でB，R，Gの3つ分を入力してください。

[P]……TPAINT(中間色ペイント)を実行します。

[SHIFT]＋[P]……PAINTを実行します。カーソルの色で境界色内を塗りつぶします。

❸ エディットコマンド類

[SHIFT]＋[B]……ポインタを先頭へ移します。

[＋]……ポインタを1つ進めます。

[－]……ポインタを1つ戻します。

[SPACE]……現在ポインタのさしているコマンドを実行します。

[／]……現在ポインタのさしているコマンドを実行し，ポインタを1つ進めます。

[INS]……コマンドのインサート。

[DEL]……コマンドのディレート。

[⇦]……現在ポインタの指しているコマンドを削除します。

[SHIFT]＋[E]……ポインタを最後に移します。

[SHIFT]＋[N]……ポインタの番号をキーボードより入力します。

[CLS]……画面をクリアします。

[SHIFT]＋[C]……メモリをクリアします。16進4桁で先頭アドレスと終了アドレスを入力してください。

[SHIFT]＋[S]……中間色のサンプルを表示します。“MC:”と表示されますのでベースになる色を入力してください。そうすると，“MC:”で入力した色とほか6色の2色の組み合わせの中間色が表示され，中央付近にグラフィックカーソルが出てきますので，タイルパターンを知りたい色の所へ移動させていき，リターンキーを押せばその色のタイルパターンが表示されます。また，カーソルを移動させずにリターンキーを押せばコマンドレベルへ戻ります。

[SHIFT]＋[D]……これも中間色の見本を表示するコマンドで，“t 1, -, t 8:”と表示し入力待ちになるので，混ぜあわせる色を8ドット分打ち込みます。そうすれば，画面中央のマス目に実際にPAINTされ，その色のタイルパターンが表示されます。

[SHIFT]＋[R]……現在ポインタの示しているコマンドから順次実行しながらトレースしていきます。[ESC]キーで一時停止，リターンキーでトレースを中断します。

[R]……これもコマンドをトレースする命令ですが[SHIFT]＋[R]の場合と違って実行中のコマンドのデータ等を表示しません。エディットしたい場所がかなり後の方にあり[SHIFT]＋[R]のトレースでは待ち時間がもったいないと思ったときに使ってください。

[SHIFT]＋[W]……実際に絵を描いてみる命令です。[SHIFT]＋[R]や[R]はBASICで処理しているのに対し，このルーチンはマシン語なのでたいへん高速です。[INS]キーをうまく使うと“マシン語モニタ”でのブレークポイントのような使い方も可能です。

[SHIFT]＋[Z]……カーソル付近を拡大表示します。線がちゃんとつながっているかどうか確かめるときなどに使います。

[SHIFT]＋[F]……コマンドをサーチする命令です。

[SHIFT]＋[L]……指定したポインタ番号間にあるPSET，CONNECT命令の表示色を変更します。

[SHIFT]＋[O]……**リスト4**の形式で，コマンドのリストを画面またはプリンタに出力します。

❹ システムコマンド類

[GRAPH]＋[S]……“PEDIT”で作成した絵のデータにマシン語のメインプログラムを付けてセーブします。

[GRAPH]＋[L]……“PEDIT”で作成したマシン語ルーチンをロードします。

[GRAPH]＋[A]……データテーブル作成の基準となる先頭のポインタのアドレスを変更します。16進4桁で入力してください。

[SHIFT]＋[H]……画面のハードコピーをとります。HARDC1ではなく，画面に表示されているのと同じような色調で出力されます。

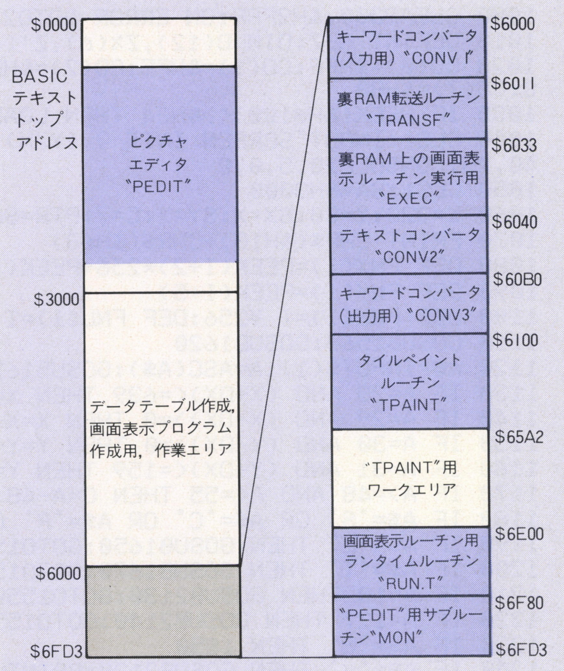
[E]……“PEDIT”の実行を終了します。

2. ピクチャエディタの使い方

まずRUNさせると，スクリーンを初期設定するかどうか聞いてきますので，以前にWIDTH 40, 25: CONSOLE 20, 5, 0, 0を実行していれば“N”，そうでなければ“Y”を入力します。そして，透明なセル，ビニールシートなどに絵を描いたものをセロテープなどで張り付け，カーソルでなぞりながら線を引いていきます。もし間違えた場合はポインタを戻して改めて正しい位置まで線を引き直します。

また線と線が閉じる所などでは拡大して確認するようにしてください。もし，色を塗ったとき色がもれてしまった場合には，一度画面をクリアして，コマンドをトレースして閉じていなかった場所を見付け出し，修正します。そして，絵を描き終わったら[GRAPH]＋[S]でセーブします。

図1　メモリマップ



キーワードコンバータ

1. テキストコンバータ "CONV2"

これはBASICのテキスト上の文字列などにオフセットを付け，判読不能にするもので，使い方は，キーワードのチェック等を行っているサブルーチンなどの前に

xxxx　'cv
　↖(適当な行番号)

という行を入れ，

UNLIST n : EXEC&H6040↵

を実行します。そうすればxxxx行以後の" "でくくられた文字列すべてにnだけオフセットがつきます。**リスト5**に使用例を示します。ただ，文字にオフセットを付けたとき，ダブルクォーテーション(CHR$ (&H22))にならぬようオフセット値を選んでください。もし，コンバートした結果，ダブルクォーテーションになるような文字があった場合には，プログラムはめちゃめちゃになります。また，CHR$ (&H1F)以下のコントロールコードにもならないよう気を付けてください。

2. キーワードコンバータ

BASICのテキスト上の文字列のみを変換しただけではあまりに片手落ちです。せっかくキーボードから入力した文字列も正しく判別されませんし，スクリーンに表示されたメッセージもおいそれとは読めないものとなってしまいます。そのため，入力した文字列にもオフセットをつけるものが"CONV1"，文字列からオフセットを取り除くためのサブルーチンが"CONV3"です。この2つもUNLIST文で指定した行番号がオフセットの値となります。**リスト6**に使用例を示します。

3. その他

"TRANSF"と"EXEC"

FM-7はF-BASICでは使われていない裏RAMをバンクレジスタ(&HFD0F)の操作で容易にアクセスすることができます。この30Kバイト以上のメモリを眠らせておく手はありません。画面データなら10枚は入ります。そこで作ったのがこの2つのプログラムです。

"TRANSF"の使い方

$600
$601 } 転送するデータの先頭アドレス
$602
$603 } 転送するバイト数
$604
$605 } 転送先のアドレス

以上のデータをセットしてEXEC &H6011で呼び出します。

"EXEC"の使い方

$600，$601番地に裏RAM上の画面データの先頭アドレスをセットし，EXEC &H6033で呼び出します。

ただし，画画表示のサブルーチンでタイルペイントを使っているときは，BIOSを裏RAM上に転送しなければなりません。ゲームを作る際，BASICのプログラム中に**リスト7**のルーチンを入れておいてください。

HARD COPY

"PEDIT"で使っているハードコピールーチンは他のサブルーチンとは無関係で$6FC6～$6FD2の部分だけあれば動きます。EXEC &H6FC6で呼び出します。

4. 参　考

リスト2の□で囲ってある所を見てください。実はこの部分がAccA(アキュムレータA)に入れたキャラクタコードにオフセットをつけているのです。リストでは，ADDA $1E8 (UNLISTフラグの下位バイト)としてあり，キャラクタコードにオフセットを加算するようになっています。これを，

	ADD	SUB	EOR
6008	BB →	BU →	B8
60A8	BB →	B0 →	B8
60BD	B0 →	BB →	B8

と変えれば，オフセットを減算したり，エクスクルーシブORをとったりするように変更できます。また，オペランドのアドレスも$1E8にこだわる必要はなく，コロコロ値が変わるようなところでなければどこの番地でも結構です。いろいろ改造して，オリジナルの方法を考えてみてください。

おわりに

今回，紙面の都合でアドベンチャゲームのサンプルを掲載できず，せっかくのユーティリティも実際にはどのように応用していくかということがわかりづらかったかもしれません。次回はアドベンチャゲームのサンプルと，具体的な作り方について説明していきたいと思います。

参考文献：Oh!FM Vol.6 p.93-99

リスト1　ピクチャエディタ・メインプログラム“PEDIT”

©1984 H・YONEMURA

デバック時は，ここに′を入れてREM文にしておくとよいでしょう。

```
1000 CLEAR300,&H2FFF:ON ERROR GOTO2290
1010 DEFINT A-Z:DIM D(12),Z%(6),Z!(6)
1020 CD(0)=&HFF:CD(1)=&HFE:CD(2)=&HEE:CD(3)=&HEA:CD(4)=&HAA:CD(5)=&HA8:CD(6)=&H8
8:CD(7)=&H80
1030 IF PEEK(&H6100)<>&H34 THEN LOADM"PD.SUB"
1040 CLS1:INPUT"SCREEN INIT ? (Y/N) ",A$:IF A$="Y" OR A$="y" THEN COLOR7,0:WIDTH
40,25:CONSOLE20,5,0,0
1050 CLS1:SA=&H3000
1060 X=320:Y=80:SX=X:SY=Y:C=7:PTR=SA:DX=1
1070 PRINT CHR$(&H1B)+CHR$(&H68)
1080 DEF FNX(I)=PEEK(I+2)*256+PEEK(I+3)
1090 DEF FNY(I)=PEEK(I+5)
1100 DEF FNH(I)=I ¥256:DEF FNL(I)=I MOD256
1110 GOSUB1560:GOSUB1620
1120 A$=INPUT$(1):A=ASC(A$):GOSUB1620
1130 IF A=28 AND (X+DX)<=639 THEN X=X+DX:GOSUB1620:GOTO1120
1140 IF A=29 AND (X-DX)>=0 THEN X=X-DX:GOSUB1620:GOTO1120
1150 IF A=30 AND (Y-DX)>=0 THEN Y=Y-DX:GOSUB1620:GOTO1120
1160 IF A=31 AND (Y+DX)<=159 THEN Y=Y+DX:GOSUB1620:GOTO1120
1170 IF A>=48 AND A<=55 THEN C=A-48:GOSUB1620:GOTO1120
1180 IF A$="S" OR A$="C" OR A$="P" OR A$="p" THEN GOSUB1640:GOTO1540
1190 IF A$="L" THEN GOSUB1650:GOTO1540
1200 IF A$="T" THEN GOSUB1670:GOTO1540
1210 IF A=18 THEN GOSUB2130:GOTO1550
1220 IF A=127 THEN GOSUB2140:GOTO1550
1230 IF A$="E" THEN 1850
1240 IF A$="a" THEN GOSUB2120:PRINTUSING"X=### Y=###";X;Y:GOTO1550
1250 IF A$="c" THEN GOSUB2150:GOTO1550
1260 IF A$="+" THEN PTR=PTR+6:GOTO1550
1270 IF A$="/" THEN GOSUB1870:PTR=PTR+6:GOTO1550
1280 IF A$=" " THEN GOSUB1870:GOTO1550
1290 IF A$="d" THEN GOSUB2690:GOTO1550
1300 IF A$="h" THEN CLS1:LINE(0,160)-(639,199),PSET,7,BF:EXEC&H6F88:CLS1:GOTO155
0
1310 IF A$="e" THEN PTR=SA:GOSUB1860:PTR=EA:GOTO1550
1320 IF A$="l" THEN GOSUB2070:GOTO1550
1330 IF A$="-" AND PTR>SA THEN PTR=PTR-6:GOTO1550
1340 IF A=8 THEN GOSUB1830:GOTO1550
1350 IF A$="n" THEN GOSUB2120:INPUT"NO·",I:PTR=SA+I*6:GOTO1550
1360 IF A$="—" THEN GOSUB2120:INPUT"addr:",D$:SA=VAL("&H"+D$):PTR=SA:GOTO1550
1370 IF A=12 THEN CLS2:GOTO1550
1380 IF A$="f" THEN GOSUB2120:PRINT"cmd:";:K=ASC(INPUT$(1)):PRINTCHR$(K):WHILE P
EEK(PTR)<>K AND PEEK(PTR)<>0:PTR=PTR+6:WEND:GOTO1550
1390 IF A$="b" THEN PTR=SA:GOTO1550
1400 IF A$="r" THEN GOSUB1900:GOTO1550
1410 IF A$="s" THEN GOSUB2360:GOTO1550
1420 IF A$="w" THEN POKE &H600,FNH(SA):POKE &H601,FNL(SA):EXEC&H6E00:GOTO1550
1430 IF A$="z" THEN GOSUB1940:GOTO1530
1440 IF A$="R" THEN GOSUB1920:GOTO1550
1450 IF A$="A" THEN GOSUB2120:INPUT"X,Y:",X,Y:GOTO1550
1460 IF A$="!" THEN DX=1:GOTO1550
1470 IF A$="(" THEN DX=8:GOTO1550
1480 IF A$="$" THEN DX=4:GOTO1550
1490 IF A=34 THEN DX=2:GOTO1550
1500 IF A$="■" THEN GOSUB2120:LINEINPUT"File name:",F$:LOADM F$:GOTO1550
1510 IF A$="|" THEN GOSUB2340
1520 IF A$="o" THEN GOSUB2580:GOTO1550
1530 GOSUB1620:GOTO1120
1540 GOSUB1690:PTR=PTR+6
1550 GOSUB1560:GOSUB1620:GOSUB2120:GOTO1120
1560 LOCATE0,20:PRINTUSING"NO·#### &  &    step # ";(PTR-SA)/6;HEX$(PTR);DX
1570 D$=CHR$(PEEK(PTR))
1580 IF D$="S" OR D$="C" OR D$="P" OR D$="p" THEN PRINTUSING"[ cmd:! ( C=# X=###
 Y=### )     ]";D$;PEEK(PTR+1);FNX(PTR);FNY(PTR):RETURN
1590 IF D$="L" THEN PRINTUSING"[ cmd:! ( L1=# L2=# L3=# L4=# )]";D$;PEEK(PTR+1);
PEEK(PTR+2),PEEK(PTR+3);PEEK(PTR+4):RETURN
1600 IF D$="T" THEN PRINTUSING"[ cmd:! ( B=&& R=&& G=&& )       ]";D$;HEX$(PEEK(PT
```

```
R+1));HEX$(PEEK(PTR+2));HEX$(PEEK(PTR+3)):RETURN
1610 PRINT"[ not used...                    ]":RETURN
1620 IF C=0 THEN LINE(X,0)-(X,159),XOR,7:LINE(0,Y)-(639,Y),XOR,7 ELSE LINE(X,0)-
(X,159),XOR,C:LINE(0,Y)-(639,Y),XOR,C
1630 LINE(SX,SY)-(X,Y),XOR,C:RETURN
1640 POKE PTR,A:POKE PTR+1,C:POKE PTR+2,FNH(X):POKE PTR+3,FNL(X):POKE PTR+4,0:PO
KE PTR+5,Y:RETURN
1650 GOSUB2120:INPUT"L1,L2,L3,L4:",L1,L2,L3,L4
1660 POKE PTR,A:POKE PTR+1,L1:POKE PTR+2,L2:POKE PTR+3,L3:POKE PTR+4,L4:POKE PTR
+5,255:RETURN
1670 GOSUB2120:INPUT"B,R,G:",B$,R$,G$:B=VAL("&H"+B$):R=VAL("&H"+R$):G=VAL("&H"+G
$)
1680 POKE PTR,A:POKE PTR+1,B:POKE PTR+2,R:POKE PTR+3,G:POKE PTR+4,255:POKE PTR+5
,255:RETURN
1690 D$=CHR$(PEEK(PTR))
1700 ON INSTR("CSpPLT",D$) GOTO1730,1720,1740,1750,1800,1810
1710 GOTO1830
1720 I=PTR:GOSUB1840:PSET(X,Y,C):SX=X:SY=Y:RETURN
1730 I=PTR:GOSUB1840:LINE(SX,SY)-(X,Y),PSET,C:SX=X:SY=Y:RETURN
1740 I=PTR:GOSUB1840:PAINT(X,Y),C,PEEK(&H50C),PEEK(&H50D),PEEK(&H50E),PEEK(&H50F
):RETURN
1750 POKE &HC5,PEEK(PTR+2):POKE &HC6,PEEK(PTR+3):POKE &HC7,0:POKE &HC8,PEEK(PTR+
5)
1760 I=PTR:GOSUB1840
1770 POKE &HF6,6:POKE &HF3,5:POKE &HF4,0
1780 POKE &H100,5:POKE &H101,PEEK(&H50C):POKE &H102,PEEK(&H50D):POKE &H103,PEEK(
&H50E):POKE &H104,PEEK(&H50F)
1790 EXEC&H6100:RETURN
1800 POKE &H50C,PEEK(PTR+1):POKE &H50D,PEEK(PTR+2):POKE &H50E,PEEK(PTR+3):POKE &
H50F,PEEK(PTR+4):RETURN
1810 POKE &H500,PEEK(PTR+1):POKE &H501,PEEK(PTR+2):POKE &H502,PEEK(PTR+3)
1820 POKE &H503,INT(PEEK(PTR+1)/8)+(PEEK(PTR+1)MOD8)*32:POKE &H504,INT(PEEK(PTR+
2)/8)+(PEEK(PTR+2)MOD8)*32:POKE &H505,INT(PEEK(PTR+3)/8)+(PEEK(PTR+3)MOD8)*32:RE
TURN
1830 FOR I=0 TO 5:POKE PTR+I,0:NEXT I:RETURN
1840 C=PEEK(I+1):X=FNX(I):Y=FNY(I):RETURN
1850 PTR=SA:GOSUB1860:CLS1:PRINT"start addr:&H";HEX$(SA):PRINT"  end addr:&H";HE
X$(EA):END
1860 POKE &H602,FNH(PTR):POKE &H603,FNL(PTR):EXEC&H6F84:EA=PEEK(&H604)*256+PEEK(
&H605):RETURN
1870 IF PEEK(PTR)=67 THEN 1890
1880 GOTO1690
1890 I=PTR-6:GOSUB1840:SX=X:SY=Y:GOTO1690
1900 WHILE PEEK(PTR)<>0 AND PEEK(&HFD01)<>13
1910 GOSUB1690:GOSUB1560:PTR=PTR+6:WEND:RETURN
1920 WHILE PEEK(PTR)<>0 AND PEEK(&HFD01)<>13
1930 GOSUB1690:PTR=PTR+6:WEND:RETURN
1940 LINE(32,20)-(39,24)," ",0,BF
1950 LINE(560,160)-(623,191),PSET,7,B
1960 LINE(588,160)-(588,191),PSET,7
1970 LINE(560,171)-(623,171),PSET,7
1980 GET@A(X-3,Y-1)-(X+4,Y+2),Z%,G
1990 FOR I=0 TO 5
2000 IF Z%(I)<0 THEN Z!(I)=65536!+Z%(I) ELSE Z!(I)=Z%(I)
2010 D(I*2)=INT(Z!(I)/256):D(I*2+1)=Z!(I)-D(I*2)*256
2020 NEXT I
2030 FOR C2=0 TO 2:FOR Y2=0 TO 3:FOR X2=0 TO 7
2040 CK=D(C2*4+Y2) AND (2^(7-X2))
2050 IF CK<>0 THEN X3=560+X2*8:Y3=160+Y2*8:LINE(X3,Y3)-(X3+7,Y3+7),OR,2^C2,BF
2060 NEXT X2,Y2,C2:RETURN
2070 GOSUB2120:INPUT"start no. , end no. :",SN,EN
2080 INPUT"C -> C2:",C2,C3
2090 FOR I=SN TO EN
2100 PTR2=SA+I*6:IF ( PEEK(PTR2)=67 OR PEEK(PTR2)=83 ) AND PEEK(PTR2+1)=C2 THEN
POKE PTR2+1,C3
2110 NEXT I:RETURN
2120 LOCATE0,22:PRINTSPACE$(32):PRINTSPACE$(32):LOCATE0,22:LINE(32,20)-(39,24),"
 ",0,BF:RETURN
2130 POKE &H602,FNH(PTR):POKE &H603,FNL(PTR):EXEC &H6F80:GOSUB1830:RETURN
2140 POKE &H602,FNH(PTR):POKE &H603,FNL(PTR):EXEC &H6F82:RETURN
2150 GOSUB2120:INPUT"start addr,end addr:",SA$,EA$:SA2=VAL("&H"+SA$):EA2=VAL("&H
```

```
"+EA$)
2160 POKE &H602,FNH(SA2):POKE &H603,FNL(SA2)
2170 POKE &H604,FNH(EA2):POKE &H605,FNL(EA2)
2180 EXEC&H6F86:RETURN
2190 POKE MA,&H30
2200 POKE MA+1,&H8D
2210 OFSET=SA-(MA+4):IF OFSET<0 THEN OF!=65536!+OFSET ELSE OF!=OFSET
2220 POKE MA+2,INT(OF!/256)
2230 POKE MA+3,OF!-INT(OF!/256)*256
2240 POKE MA+4,&HBD
2250 POKE MA+5,&H6E
2260 POKE MA+6,&H03
2270 POKE MA+7,&H39
2280 RETURN
2290 IF ERR=72 THEN GOSUB2120:COLOR2:BEEP:PRINT"Drive Not Ready !!":COLOR7:RESUM
E1550
2300 IF ERR=63 THEN GOSUB2120:COLOR2:BEEP:PRINT"File Not Found !!":COLOR7:RESUME
1550
2310 IF ERR<>64 THEN GOSUB2120:BEEP:COLOR2:PRINT"ERROR";ERR;"IN";ERL:COLOR7:END
2320 GOSUB2120:BEEP:COLOR2:PRINT"File Already Exists !!":PRINT"Are you sure ? (Y
/N)";:D$=INPUT$(1):PRINTD$:COLOR7:IF D$="Y" OR D$="y" THEN KILL F$:SAVEM F$,SA,E
A,MA
2330 RESUME1550
2340 GOSUB2120:GOSUB1860:MA=EA+6:GOSUB2190:EA=MA+8
2350 PRINTHEX$(SA);",";HEX$(EA);",";HEX$(MA):LINEINPUT"File name:",F$:IF F$<>""
THEN SAVEM F$,SA,EA,MA:RETURN ELSE RETURN
2360 CLS:LINE(0,0)-(639,159),PSET,7,B
2370 INPUT"MC:",MC
2380 FOR X=0 TO 639 STEP80:LINE(X,0)-(X,159),PSET,7:NEXT X
2390 IF MC>=6 THEN PC=MC-1 ELSE PC=MC+1
2400 PAINT(40,40),PC,7
2410 IF MC=0 THEN FOR X=200 TO 639 STEP80:PAINT(X,40),1,7:NEXT X:PAINT(120,40),2
,7
2420 FOR Y=0 TO 159 STEP20:LINE(0,Y)-(639,Y),PSET,7:NEXT Y
2430 FOR I=0 TO 3:POKE &H50C+I,7:NEXT I
2440 FOR X=0 TO 7:FOR Y=0 TO 7
2450 GOSUB2540
2460 X2=X*80+40:Y2=Y*20+1
2470 POKE &HC5,FNH(X2):POKE &HC6,FNL(X2):POKE &HC7,0:POKE &HC8,Y2
2480 POKE &H500,B:POKE &H501,R:POKE &H502,G
2490 POKE &H503,INT(B/8)+(B MOD8)*32:POKE &H504,INT(R/8)+(R MOD8)*32:POKE &H505,
INT(G/8)+(G MOD8)*32
2500 GOSUB1770:NEXT Y,X:X=328:Y=84
2510 GCURSOR(X,Y),(X2,Y2)
2520 IF X2=X AND Y2=Y THEN CLS:RETURN
2530 X=INT(X2/80):Y=INT(Y2/20):GOSUB2540:PRINTUSING"B=&& R=&& G=&&";HEX$(B);HEX$
(R);HEX$(G):X=X2:Y=Y2:GOTO2510
2540 B=(((X AND1)<>0)*(-CD(Y)))OR(((MC AND1)<>0)*(CD(Y)-255))
2550 R=(((X AND2)<>0)*(-CD(Y)))OR(((MC AND2)<>0)*(CD(Y)-255))
2560 G=(((X AND4)<>0)*(-CD(Y)))OR(((MC AND4)<>0)*(CD(Y)-255))
2570 RETURN
2580 GOSUB2120:PRINT"SCRN: or LPT0: (S/L) ";:O$=INPUT$(1):PRINTO$
2590 IF O$="L" THEN OPEN"O",#1,"LPT0:" ELSE OPEN"O",#1,"SCRN:":WIDTH80,25
2600 I=PTR:WHILE PEEK(I)<>0 AND PEEK(&HFD01)<>13
2610 D$=CHR$(PEEK(I))
2620 IF D$="L" THEN PRINT#1:PRINT#1,USING"[####] limit(# # # #)";(I-SA)/6;PEEK(I
+1);PEEK(I+2);PEEK(I+3);PEEK(I+4);
2630 IF D$="T" THEN PRINT#1:PRINT#1,USING"[####] tile(&& && &&)";(I-SA)/6;HEX$(P
EEK(I+1));HEX$(PEEK(I+2));HEX$(PEEK(I+3));
2640 IF D$="P" THEN PRINT#1:PRINT#1,USING"[####] tpaint(### ###)";(I-SA)/6;FNX(I
);FNY(I);
2650 IF D$="p" THEN PRINT#1:PRINT#1,USING"[####] paint(### ### #)";(I-SA)/6;FNX(
I);FNY(I);PEEK(I+1);
2660 IF D$="S" THEN PRINT#1:PRINT#1,USING"[####] pset(### ### #) connect";(I-SA)
/6;FNX(I);FNY(I);PEEK(I+1);
2670 IF D$="C" THEN PRINT#1,USING"-(### ### #)";FNX(I);FNY(I);PEEK(I+1);
2680 I=I+6:WEND:CLOSE:IF O$<>"L" THEN WIDTH40,25:CONSOLE20,5,0,0:RETURN ELSE RET
URN
2690 CLS2:LINE(280,60)-(360,100),PSET,7,B
2700 INPUT"t1,-,t8:",T(7),T(6),T(5),T(4),T(3),T(2),T(1),T(0)
2710 PC=0:FOR I=0 TO 7:IF T(I)=PC THEN PC=PC+1
```

```
2720 NEXT I:PAINT(320,80),PC,7
2730 B=0:R=0:G=0:FOR I=0 TO 7:B=B+(T(I) AND 1)*2^I:R=R+(T(I) AND 2)*2^(I-1):G=G+
(T(I) AND 4)*2^(I-2):NEXT I
2740 FOR I=0 TO 3:POKE &H50C+I,7:NEXT I
2750 POKE &HC5,1:POKE &HC6,64:POKE &HC7,0:POKE &HC8,80
2760 POKE &H500,B:POKE &H501,R:POKE &H502,G
2770 POKE &H503,INT(B/8)+(B MOD8)*32:POKE &H504,INT(R/8)+(R MOD8)*32:POKE &H505,
INT(G/8)+(G MOD8)*32
2780 GOSUB1770:PRINTUSING"B=&& R=&& G=&&";HEX$(B);HEX$(R);HEX$(G):RETURN
```

リスト2　キーワードコンバータ，テキストコンバータ，タイルペイントルーチン

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6000 E6 84 AE 01 30 1F A6 85 93 ▀ｮ.0.ｦ■
6008 BB 01 E8 A7 85 5A 26 F6 46 ｻ.♠ｧ■Z&分
6010 39 34 60 B7 FD 0F BE 06 54 94`ｷ .ｾ.
6018 00 10 BE 06 04 CC 00 00 A4 ..ｾ..ﾌ..
6020 EE 8B EF AB C3 00 02 10 E8 ◢▌╲ｫﾃ...
6028 B3 06 02 23 F3 B6 FD 0F 93 ｳ..#月ｶ .
6030 35 60 39 BE 06 00 B7 FD 46 5`9ｾ..ｷ
6038 0F 17 0D C7 B6 FD 0F 39 F5 ...ﾇｶ .9
6040 6F 8D 00 63 9E 33 30 03 63 o▌.c└30.
6048 30 01 A6 84 81 8D 27 3F CF 0.ｦ▄▁▌'?
6050 6D 8D 00 53 27 08 81 FE FB m▌.S'.▁
6058 27 13 81 22 27 1F 81 00 A4 '.▁"'.▁.
6060 26 E6 30 04 30 84 9C 35 C5 &◥0.0▄╭5
6068 24 02 20 DC 39 30 01 A6 32 $. ﾜ90.ｦ
6070 84 81 F2 27 04 30 86 20 F8 ▄▁年'.0■
6078 CF 30 02 20 CB 30 01 A6 C3 ﾏ0. ﾋ0.ｦ
------------------------------------------
SUM  8F 98 56 3B CD 02 CC B7 0A
```

FM-8はD352

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6080 84 81 22 27 C3 BD D9 D9 80 ▄▁"'ﾃｽﾙﾙ
6088 BB 01 E8 A7 84 20 EE 30 0D ｻ.♠ｧ▄ ◢0
6090 01 A6 84 81 63 26 C7 30 2C .ｦ▄▁c&ﾇ0
6098 01 A6 84 81 76 26 BF 86 8D .ｦ▄▁v&ｿ■
60A0 01 A7 8D 00 02 20 B7 00 0E .ｧ▌.. ｷ.
60A8 BB 01 E8 A7 85 20 F0 00 E0 ｻ.♠ｧ■ ×.
60B0 E6 84 F7 60 F7 C6 01 AE 2D ◥▄秒`秒ﾆ.ｮ
60B8 01 30 1F A6 85 B0 01 E8 14 .0.ｦ■ｰ.♠
60C0 B7 60 F6 86 81 B7 FD 03 CB ｷ`分■▁ｷ .
60C8 34 14 30 8D 00 1E AD 9F 6F 4.0▌..ｭ╯
60D0 FB FA 8E 3F FF 86 01 B7 FF  ▌?.■.ｷ
60D8 FD 03 30 1F 26 FC 35 14 BA  .0.& 5.
60E0 5C F1 60 F7 23 D5 0F BF 6A ¥円`秒#ﾕ.ｿ
60E8 BD 9B 50 39 10 00 60 F2 43 ｽ┘P9..`年
60F0 00 10 00 00 03 01 46 03 5D ......F.
60F8 B7 FD 05 39 81 08 26 F3 94 ｷ .9▁.&月
------------------------------------------
SUM  97 34 36 57 80 14 B1 69 06
```

FM-8は9F97

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6100 34 7F 8E 05 A0 CE 10 00 C4 4.▌..ﾎ..
6108 EF 84 33 8D 00 0B EF 02 2F ╲▄3▌..╲.
6110 CE 00 80 EF 04 9D DE 20 DC ﾎ.▁╲.╮ﾞ
6118 7B 00 00 3F 59 41 4D 41 E2 {..?YAMA
6120 55 43 48 49 93 D3 96 91 B6 UCHI├ﾓ│┬
6128 D3 9A 3E 80 00 66 90 B6 D7 ﾓ└>▁.f┴ｶ
6130 D4 09 39 BE D3 92 B6 D3 C2 ﾔ.9ｾﾓ┤ｶﾓ
6138 94 B7 D3 A1 BC D3 95 27 0A ▔ｷﾓ｡ｼﾓ─'
6140 24 8D 2B 7F D3 A1 7A D3 1C $▌+.ﾓ｡zﾓ
6148 A1 30 01 BC D3 95 27 15 32 ｡0.ｼﾓ─'.
6150 F6 D3 8F E7 84 F6 D3 90 1C 分ﾓ┼◤▄分ﾓ┴
6158 E7 89 40 00 F6 D3 91 E7 F1 ◤▎@.分ﾓ┬◤
6160 89 80 00 20 E4 F6 D3 97 6D ▎▁. ◢分ﾓ│
6168 F4 D3 A1 F7 D3 A1 10 8E 71 日ﾓ｡秒ﾓ｡.■
6170 D3 8F 5F B6 D3 A1 A4 A5 34 ﾓ┼_ｶﾓ｡､･
6178 B7 D3 A2 B6 D3 A1 43 A4 3D ｷﾓ｢ｶﾓ｡C､
------------------------------------------
SUM  A5 6E 70 8D 9C 2D 6A 71 B4
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6180 84 BA D3 A2 A7 84 30 89 97 ▄ｺﾓ｢ｧ▄0▎
6188 40 00 5C C1 03 26 E4 30 9A @.¥ﾁ.&◢0
6190 89 40 00 39 8E 05 A0 CE 03 ▎@.9■..ﾎ
6198 10 00 EF 84 33 8D 00 0B 4E ..╲▄3▌..
61A0 EF 02 CE 00 80 EF 04 9D CF ╲.ﾎ.▁╲.╮
61A8 DE 20 62 00 00 3F 59 41 39 ﾞ b..?YA
61B0 4D 41 55 43 48 49 93 D3 1D MAUCHI├ﾓ
61B8 96 91 D3 9A 3F 00 00 66 39 │┬ﾓ└?..f
61C0 90 B6 D4 09 39 BE D3 92 7F ┴ｶﾔ.9ｾﾓ┤
61C8 BD 3F 80 B4 D3 94 20 0D C4 ｽ?▁ｴﾓ▔ .
61D0 BC D3 95 27 15 B6 D3 A5 8E ｼﾓ─'.ｶﾓ･
61D8 30 86 BD 3F 80 BC D3 92 53 0■ｽ?▁ｼﾓ┤
61E0 26 03 B4 D3 97 4D 27 E8 A3 &.ｴﾓ M'♠
61E8 20 04 B6 D3 97 43 BF D3 19  .ｶﾓ Cｿﾓ
61F0 A1 5F 53 7D D3 A5 2B 07 7A ｡_S}ﾓ･+.
61F8 56 49 24 FC 53 20 04 59 8F VI$ S .Y
------------------------------------------
SUM  83 EB FD 3F 67 CC 52 9A C9
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6200 46 24 FC F7 D3 A3 53 B6 DC F$ 秒ﾓ｣Sｶ
6208 D3 A2 1F 03 39 8E 05 A0 03 ﾓ｢..9■..
6210 CE 10 00 EF 84 33 8D 00 11 ﾎ..╲▄3▌.
6218 0B EF 02 CE 00 80 EF 04 3D .╲.ﾎ.▁╲.
6220 9D DE 20 7A 00 00 3F 59 AD ╮ﾞ z..?Y
6228 41 4D 41 55 43 48 49 93 8B AMAUCHI├
6230 D3 96 91 D3 9A 3F 80 00 26 ﾓ│┬ﾓ└?▁.
6238 66 90 B6 D4 09 39 CE D3 63 f┴ｶﾔ.9ﾎﾓ
6240 8F 10 8E D3 98 7F D3 A1 8B ┼.■ﾓ┌.ﾓ｡
6248 C6 03 A6 C0 A8 84 BA D3 E8 ﾆ.ｦﾀｨ▄ｺﾓ
6250 A1 B7 D3 A1 30 89 40 00 C5 ｡ｷﾓ｡0▎@.
6258 5A 26 EF 30 89 40 00 B6 1E Z&╲0▎@.ｶ
6260 D3 98 A7 8D 00 12 6A 8D A8 ﾓ┌ｧ▌..j▌
6268 00 0E 27 2B 7F D3 A2 C6 1A ..'+.ﾓ｢ﾆ
6270 04 30 89 80 00 4F E5 A8 19 .0▎▁.O◣ｨ
6278 40 27 01 4A A8 84 BA D3 6B @'.Jｨ▄ｺﾓ
------------------------------------------
SUM  70 03 13 13 96 28 22 11 8A
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6280 A2 B7 D3 A2 30 89 C0 00 47 ｢ｷﾓ｢0▎ﾀ.
6288 54 26 EA 30 89 40 00 B4 11 T&◆0▎@.ｴ
6290 D3 A1 B7 D3 A1 20 CF B6 44 ﾓ｡ｷﾓ｡ ﾏｶ
6298 D3 A1 B8 D3 A4 39 8E 05 6F ﾓ｡ｸﾓ､9■.
62A0 A0 CE 10 00 EF 84 33 8D B1 .ﾎ..╲▄3▌
62A8 00 0B EF 02 CE 00 80 EF 39 ..╲.ﾎ.▁╲
62B0 04 9D DE 20 7D 00 00 3F 5B .╮ﾞ }..?
62B8 59 41 4D 41 55 43 48 49 51 YAMAUCHI
62C0 93 D3 96 91 D3 9A 7E 80 F8 ├ﾓ│┬ﾓ└~▁
62C8 00 66 90 B6 D4 09 39 70 32 .f┴ｶﾔ.9p
62D0 D3 A5 BD 3F 00 70 D3 A5 5C ﾓ･ｽ?.pﾓ･
62D8 FC D3 95 C3 00 4F FD D3 46  ﾓ─ﾃ.O ﾓ
62E0 95 FC D3 93 B3 D3 A2 26 45 ─ ﾓ├ｳﾓ｢&
62E8 08 8D 2A 25 04 73 D3 92 C0 .▌*%.sﾓ┤
62F0 39 BF D3 92 FF D3 93 BF 81 9ｿﾓ┤.ﾓ├ｿ
62F8 D3 B0 FF D3 B1 BD 3F 00 02 ﾓｰ.ﾓｱｽ?.
------------------------------------------
SUM  A4 7F 9D 41 9B 21 E6 52 F5
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6300 8D 13 24 03 73 D3 A1 BE 6C ▌.$.sﾓ｡ｾ
6308 D3 92 B6 D3 94 43 BF D3 57 ﾓ┤ｶﾓ▔Cｿﾓ
6310 95 B7 D3 97 39 BF D3 92 13 ─ｷﾓ 9ｿﾓ┤
6318 FF D3 93 73 D3 A4 BD 3F 4B .ﾓ├sﾓ､ｽ?
6320 00 73 D3 A4 BC D3 AD 25 4B .sﾓ､ｼﾓｭ%
```

```
6328 08 26 06 B6 D3 A3 B1 D3 E4 .&.カモ」アモ
6330 AF 39 30 8D 00 3B CE 01 AF ッ90■.;ホ.
6338 00 86 09 E6 C0 E7 80 4A E6 .■.◥タ◤_」
6340 26 F9 8E 05 A0 CE 10 00 30 & ■..ホ..
6348 EF 84 33 8D 00 0B EF 02 2F ＼▃3■..＼.
6350 CE 00 80 EF 04 9D DE 20 DC ホ._＼.ヽ゛
6358 2D 00 00 3F 59 41 4D 41 94 -..?YAMA
6360 55 43 48 49 93 D3 A6 90 C5 UCHI┣モヲ┴
6368 55 FF 55 00 00 00 00 00 A9 U.U.....
6370 00 05 07 07 02 00 01 40 56 .......@
6378 00 50 00 00 00 FF 01 B6 06 .P.....カ
--------------------------------------------
SUM  65 9B 37 BD F4 9A 6E 8E 7E
```

FM-8は97 40

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6380 D4 09 BD 3F 80 39 96 F6 1E ヤ.ス?_9┃分
6388 5F 5C 80 03 24 FB 5A 26 DD _¥_.$ Z&
6390 03 7E 92 A0 D7 F6 DC C5 21 .~┤.ラ分ワナ
6398 44 56 44 56 44 56 1F 98 85 DVDVDV.┌
63A0 1F 01 D6 C6 C4 07 D7 C6 24 ..ヨニト.ラニ
63A8 58 58 58 DA C6 96 C8 10 16 XXXレニ┃ネ.
63B0 EF 8D 01 EE 32 8D 09 EA 1D ◣■./2■.◆
63B8 33 8D 01 E8 10 8E FE FE 43 3■.♠.■
63C0 34 20 34 20 10 AF C1 10 38 4 4 .ッチ.
63C8 AF C1 34 16 CA 40 4A 34 42 ッチ4.ハ@J4
63D0 16 0F F5 7D 03 13 27 07 DB ..時}..´.
63D8 10 EE 8D 01 C5 35 FF 1F A4 ./■.ナ5..
63E0 30 1F 41 43 53 30 8B 8C 6D 0.ACS0▌▌
63E8 00 40 25 20 0D F5 26 04 B1 .@% .時&.
63F0 35 16 20 04 AE C3 EC C3 8F 5. .ョテ●テ
63F8 81 FE 26 1F 0D F5 27 04 F1 ▁ &..時´.
--------------------------------------------
SUM  02 FD D9 E8 48 4C 86 F8 D2
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6400 35 16 20 04 AE C3 EC C3 8F 5. .ョテ●テ
6408 81 FE 26 07 10 EE 8D 01 38 ▁ &../■.
6410 91 35 FF 33 44 32 7C 03 ED ┬5.3D2¦.
6418 F5 20 B8 97 C8 1F 98 C4 A7 時 ク ┃ネ.┌ト
6420 07 D7 F2 1F 89 C4 C0 D7 D3 .ラ年.▌トタラ
6428 C7 84 38 44 44 44 97 F0 D6 ヌ▄8DDD ┃╳
6430 1F 10 97 EF D7 F1 86 40 43 .. ┃◣ラ円■@
6438 95 C7 26 14 96 C8 80 01 75 ─ヌ&.┃ネ_.
6440 25 08 97 C6 8D 7D 0D F7 98 %. ┃ニ■}.秒
6448 2A 1C 86 40 9A C7 97 C7 CB *.■@└ヌ ┃ヒ
6450 0D C7 2B 0F 96 C8 4C 81 39 .ヌ+.┃ネL▁
6458 C8 24 08 97 C6 8D 64 0D 4F ネ$. ┃ニ■d.
6460 F7 2A 03 16 FF 6D 86 C0 EC 秒*...m■タ
6468 90 C5 27 1D 96 EF D6 F1 E5 ┴ナ´.┃◣ヨ円
6470 1F 01 D6 F0 58 58 58 DA C8 ..ヨ╳XXXレ
6478 F2 DA C5 96 C8 0D F5 26 17 年レナ┃ネ.時&
--------------------------------------------
SUM  7A 74 F9 A0 3C 1D E7 90 57
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6480 04 34 16 20 04 ED C1 AF CF .4. .○チッ
6488 C1 96 F7 D6 F9 1F 01 D6 13 チ┃秒ヨ ..ヨ
6490 F8 58 58 58 DA FA 4F 10 33  XXXレ O.
6498 9E EF 10 9C F7 22 11 10 73 ╲◣.╭秒"..
64A0 9E F1 10 9C F9 25 09 43 A5 ╲円.╭ %.C
64A8 0D C7 26 02 CA 80 DA C7 E7 .ヌ&.ハ_レヌ
64B0 98 F5 27 06 96 C6 34 16 60 ┌時´.┃ニ4.
64B8 20 06 96 C6 ED C1 AF C1 A0  .┃ニ○チッチ
64C0 16 FF 10 17 00 B4 96 C6 4C .....エ┃ニ
64C8 C6 50 3D DD F9 FD FC 95 B7 ニP=ン    ─
64D0 86 FF B7 FC 97 8E FC 92 EB ■.ｷ  ┃■ ┤
64D8 10 8E 00 EF 17 00 B0 8E E2 .■.◣..─■
64E0 FC AD 17 00 AA 8E 7E 80 F6  ュ..ェ■~_
64E8 BF FC AA 86 3F B7 FC 82 5F ソ ェ■?ｷ ▂
64F0 96 C6 90 F6 24 FC 9B F6 93 ┃ニ┴分$ ┘分
64F8 C6 03 3D D3 F3 1F 01 EC D8 ニ.=モ月..●
--------------------------------------------
SUM  47 12 FA 82 BB F3 3C E5 A4
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6500 81 FD FC 8F A6 84 B7 FC E6 ▁  ┼ヲ▄ｷ
6508 91 86 00 B7 FD 05 4A 26 40 ┬■.ｷ .J&
6510 FD 8D 67 0F F7 7D FC 92 02  ■g.秒} ┤
6518 2B 53 FC FC B0 FD FC 92 B1 +S  ─  ┤
6520 93 F9 D7 F7 F6 FC B2 F7 F5 ┣ ラ秒分 イ秒
6528 FC 94 86 09 4A 54 25 FC DE  ▔■.JT%
6530 97 F8 81 08 26 04 0F F8 49 ┃ ▁.&..
6538 0C F7 FC FC 95 93 F9 D7 F3 .秒  ─┣ ラ
6540 F9 F6 FC 97 86 FE 4C 58 AA  分 ┃■ LX
6548 25 FC 97 FA 81 FF 26 06 5E % ┃ ▁.&.
6550 86 07 97 FA 0A F9 96 C7 7E ■.┃ . ┃ヌ
6558 7D FC A1 2B 02 8B 40 9A AC }  。+.▌@└
6560 C7 97 C5 8E 3E 80 BF FC 2A ヌ┃ナ■>_ソ
6568 AA 86 3F 20 04 03 F7 86 13 ェ■? ..秒■
6570 80 B7 FC 82 86 00 B7 FD EF _ｷ ▂■.ｷ
6578 05 39 B6 FD 05 2B FB 86 A2 .9カ .+ ■
--------------------------------------------
SUM  83 E1 BA 38 25 19 88 CC E8
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6580 80 B7 FD 05 C6 08 5A 27 88 _ｷ .ニ.Z´
6588 05 B6 FD 05 2A F8 39 E6 FE .カ .* 9◥
6590 21 4F 43 44 5A 2A FC 43 BA !OCDZ* C
6598 A7 02 4F E6 A1 D3 F9 ED 38 ァ.O◥。モ ○
65A0 84 39 00 00 00 00 00 00 BD ▄9......
--------------------------------------------
SUM  D1 F7 8C 34 EB FD 88 3D 35
```

リスト 3　画面表示ランタイムルーチン，“PEDIT”サブルーチン

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6E00 BE 06 00 A6 84 81 54 10 D3 セ..ヲ▄▁T.
6E08 27 00 95 81 50 10 27 00 C4 ´.─▁P.´.
6E10 D7 81 4C 27 75 81 43 27 2B ラ▁L´u▁C´
6E18 0B 81 53 27 41 81 70 10 48 .▁S´A▁p.
6E20 27 00 F9 39 17 01 38 86 2F ´. 9..8■
6E28 15 B7 FC 82 A6 01 B7 FC A4 .ｷ ▂ヲ.ｷ
6E30 83 86 00 B7 FC 84 FC 6F AB ▃■.ｷ ▄ o
6E38 54 FD FC 85 FC 6F 56 FD 90 T  ■ oV
6E40 FC 87 EC 02 FD FC 89 FD F0  ▇●.  ▌
6E48 6F 54 EC 04 FD FC 8B FD 34 oT●.  ▌
6E50 6F 56 86 00 B7 FC 8D 17 A2 oV■.ｷ ■.
6E58 01 15 30 06 20 A5 17 00 28 ..0. ・..
6E60 FE 86 17 B7 FC 82 86 01 57  ■.ｷ ▂■.
6E68 B7 FC 83 EC 02 FD FC 84 A1 ｷ ▃●.  ▄
6E70 FD 6F 54 EC 04 FD FC 86 2F  oT●.  ■
6E78 FD 6F 56 A6 01 B7 FC 88 A4  oVヲ.ｷ ▏
--------------------------------------------
SUM  64 E8 F7 AD 13 54 A1 D9 D1
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6E80 86 00 B7 FC 89 17 00 E7 C0 ■.ｷ ▌..◤
6E88 20 D0 A6 01 B7 05 0C A6 05  ミヲ.ｷ..ヲ
6E90 02 B7 05 0D A6 03 B7 05 30 .ｷ..ヲ.ｷ.
6E98 0E A6 04 B7 05 0F 20 BA 5D .ヲ.ｷ.. コ
6EA0 A6 01 B7 05 00 48 89 00 34 ヲ.ｷ..H▌.
6EA8 B7 05 03 48 89 00 B7 05 4C ｷ..H▌.ｷ.
6EB0 06 48 89 00 B7 05 09 A6 42 .H▌.ｷ..ヲ
6EB8 02 B7 05 01 48 89 00 B7 47 .ｷ..H▌.ｷ
6EC0 05 04 48 89 00 B7 05 07 9D ..H▌.ｷ..
6EC8 48 89 00 B7 05 0A A6 03 40 H▌.ｷ..ヲ.
6ED0 B7 05 02 48 89 00 B7 05 4B ｷ..H▌.ｷ.
6ED8 05 48 89 00 B7 05 08 48 E2 .H▌.ｷ..H
6EE0 89 00 B7 05 0B 16 FF 72 D7 ▌.ｷ....r
6EE8 EC 02 DD C5 EC 04 DD C7 24 ●.ンナ●.ンヌ
6EF0 86 0C 97 F6 CC 05 00 DD CD ■.┃分フ..ン
6EF8 F3 86 05 B7 01 00 B6 05 F1 月■.ｷ..カ.
--------------------------------------------
SUM  12 A0 B1 0E 7C E9 28 20 1E
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6F00 0C B7 01 01 B6 05 0D B7 44 .ｷ..ｶ..ｷ
6F08 01 02 B6 05 0E B7 01 03 87 ..ｶ..ｷ..
6F10 B6 05 0F B7 01 04 BD 61 A4 ｶ..ｷ..ｽa
6F18 00 16 FF 3E 8D 41 86 18 BF ...>■A■.
6F20 B7 FC 82 EC 02 FD FC 83 9F ｷ ▂●.  ▬
6F28 EC 04 FD FC 85 A6 01 B7 CC ●.  ■ｦ.ｷ
6F30 FC 87 86 04 B7 FC 88 B6 FE  ■■.ｷ ▎ｶ
6F38 05 0C B7 FC 89 B6 05 0D 15 ..ｷ ▎ｶ..
6F40 B7 FC 8A B6 05 0E B7 FC B9 ｷ ▎ｶ..ｷ
6F48 8B B6 05 0F B7 FC 8C 8D 21 ▌ｶ..ｷ ▐■
6F50 1E 16 FF 06 01 43 00 13 90 .....C..
6F58 00 00 00 00 FF 11 11 B6 D7 .......ｶ
6F60 FD 05 2B FB 86 80 B7 FD E2  .+ ■▁ｷ
6F68 05 B6 FD 05 2A FB 39 86 A1 .ｶ .* 9■
6F70 00 B7 FD 05 39 00 00 00 F2 .ｷ .9...
6F78 00 00 00 00 00 00 00 34 34 .......4
-----------------------------------------
SUM  C9 A1 34 B3 BE 2F 1F 39 96
```

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
6F80 20 08 20 17 20 26 20 33 F8  . . & 3
6F88 20 3C 8D 20 BE 06 04 A6 77  <■ ｾ..ｦ
6F90 84 A7 06 30 1F BC 06 02 44 ▄ｧ.0.ｼ..
6F98 24 F5 39 8D 0F BE 06 02 B4 $■9■.ｾ..
6FA0 A6 06 A7 84 30 01 BC 06 CA ｦ.ｧ▄0.ｼ.
6FA8 04 23 F5 39 BE 06 02 30 4B .#■9ｾ..0
6FB0 1A 30 06 A6 84 26 FA BF 59 .0.ｦ▄& ｿ
6FB8 06 04 39 BE 06 02 6F 80 F8 ..9ｾ..o▁
6FC0 BC 06 04 23 F9 39 8E 6F 18 ｼ..# 9■o
6FC8 CE AD 9F FB FA 39 0F 00 57 ﾎｭ   9..
6FD0 6D 00 07 38 00 00 00 00 AC m..8....
6FD8 00 00 00 00 00 00 00 00 00 ........
6FE0 00 00 00 00 00 00 00 00 00 ........
6FE8 00 00 00 00 00 00 00 00 00 ........
6FF0 00 00 00 00 00 00 00 00 00 ........
6FF8 00 00 00 00 00 00 00 00 00 ........
-----------------------------------------
SUM  A9 F0 71 6B 77 47 F4 C1 E8
```

リスト 4

```
[    0] pset(  0    0 7) connect-(  0 159 7)-(639 159 7)-(639    0 7)-(  0    0 7)
[    5] limit(7 7 7 7)
[    6] paint(   1    1 5)
[    7] pset(   1   33 0) connect-(113   33 0)-(174   45 0)-(467   45 0)-(514   33 0)-(
638   33 0)
[   13] pset(638 151 0) connect-(518 151 0)-(472 139 0)-(175 139 0)-(110 152 0)-(
  1 152 0)
```

リスト 5

```
10 A$="ﾀﾀﾞﾉ TEST ﾃﾞｽﾖｰ!"
20 IF INSTR(A$,"TEST") THEN PRINT"ﾀﾀﾞﾉ SAMPLE ﾃﾞｽﾖｰ!"
30 'cv
40 A$="ﾀﾀﾞﾉ TEST ﾃﾞｽﾖｰ!"
50 IF INSTR(A$,"TEST") THEN PRINT"ﾀﾀﾞﾉ SAMPLE ﾃﾞｽﾖｰ!"

UNLIST 5:EXEC&H6040
```

これで下記のようなリストになりますが，LIST は見ることができません。POKE &HIE7, &HFF で確認できます。

```
10 A$="ﾀﾀﾞﾉ TEST ﾃﾞｽﾖｰ!"
20 IF INSTR(A$,"TEST") THEN PRINT"ﾀﾀﾞﾉ SAMPLE ﾃﾞｽﾖｰ!"
30 'cv
40 A$="ﾅﾅｮﾎ%YJXY%ﾈｮﾂﾛｵ&"
50 IF INSTR(A$,"YJXY") THEN PRINT"ﾅﾅｮﾎ%XFRUQJ%ﾈｮﾂﾛｵ&"
```

リスト 6

```
10 UNLIST 5:DELETE 10
20 DEF USR0=&H6000:DEF USR1=&H60B0
30 LINEINPUT"▶ ",A$:PRINT
40 PRINT "conv1    ";A$;" -> ";USR0(A$)
50 B$="ﾅﾅｮﾎ%YJXY%ﾈｮﾂﾛｵ&"
60 PRINT "conv3    ";B$;" -> ";:B$=USR1(B$)
```

リスト 7

```
1000 FOR I=&HF17D TO &HFBFF:POKE I-&HA000,PEEK(I):NEXT
1010 POKE &H600,&H50:POKE &H601,&H00:POKE &H602,&H0B:POKE &H603,&HFF:POKE &H604,
&HF0:POKE &H605,&H00:EXEC&H6011
```

リスト 1 のBASICチェックサム(p. 41参照)

```
LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
1000 FF EE 44 D1 82 4F 8E 19 :7A
1080 BD B2 4A CF A0 72 07 02 :A3
1160 6C F1 07 35 3F 29 5B 73 :CF
1240 D5 49 53 E1 0E 53 33 46 :2C
1320 53 F2 FE A7 EC F1 70 C7 :FE
1400 5A 5C 3C 64 3C 0E C9 D7 :40
1480 CF EC 85 3F 5C 80 14 85 :F4
1560 E6 63 44 74 0F 6B 85 05 :05
1640 13 71 CF 35 4C 63 DE 05 :1A
1720 26 7B 56 47 03 D9 1F 74 :AD
---------------------------------
     98 63 10 F0 51 63 F2 75 /16

LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
1800 B8 EC 16 80 61 EB 02 A6 :2E
1880 09 9F 0F 1C 0F 96 44 F0 :AC
1960 96 82 82 75 97 E4 88 7F :91
2040 DC 4F 3B 1C 70 44 19 A2 :F1
2120 DC 98 14 72 E7 CF 97 BA :01
2200 2F 08 04 AF 3C 32 1B 25 :98
2280 E0 22 AF 77 7E 9C 2E 9B :0B
2360 45 16 39 A2 E3 1D 37 8B :F8
2440 47 4B 83 E5 6C A2 4D 11 :66
2520 CD 2B 98 AA A3 E0 EB BC :64
---------------------------------
     77 AA FD F6 0A E5 36 89 /C2

LINE +0 +1 +2 +3 +4 +5 +6 +7 SUM
2600 18 B6 B1 35 51 96 23 83 :41
2680 DB 00 C4 26 DA DA 8B EB :EF
2760 6C A2 06 00 00 00 00 00 :14
---------------------------------
     5F 58 7B 5B 2B 70 AE 6E /44
```

続・BIOSの使い方［2］

for FM-7, 8, 11

――初歩の機械語入門

古田 昌

「BIOSの使い方」も3回目を迎え，いよいよ佳境に入ってきました。前回では，スピードゲーム用BIOSの使い方として，出力プログラムの話でしたが，今回はその入力プログラム編として，SUBIN，KEYIN，ANALGPの3つの命令について説明します。ゲームへの応用例については，次回でふれたいと思います。

入力プログラム

⑤ SUBIN編

まず，BIOS-SUBINの出番です。サブシステムにデータを転送したあと，サブシステムからのデータの入力が必要なときに用いられるのが，このSUBINです。これもHLT(ホールト)，MOVE(ムーブ)をBIOSがやってくれる便利なコマンドです。

図2―1が実行前のダンプリストです。ここでは，タイマレジスタの読み込みを例にとっていますから，実行後のダンプリストである**図2―2**の値は不定です。

例によって「打ち込み，セーブ，実行」のあと「ロード，デバッグ，セーブ，実行」を繰り返してください。

それでは**図2―3**の解説にいきます。

まずRCBの説明です。リクエスト番号は17（$11）です。RCBSTAは1バイト必要で，サブシステムのエラーがあったときは，ここにエラー番号が書き込まれます。データ領域番地（RCBDBA）をDATAのアドレスにして，転送データ長RCBLNHを128にします。BIOSリターン時には，RCBLNHに復帰情報のバイト数が書き込まれています。RCBBMHは，データバッファへの最大入力可能バイト数で，ユーザがデータ領域の長さに対してセットします。サブシステムからのデータがこれを超えて転送されることはありません。

データの転送についてふれておきましょう。**図2―4**を見てください。

SUBINは，サブCPUを待ち状態（HLT）にしたあと，RCBLNHのバイト数だけデータ領域のデータを中間領域に書き込み，サブCPUを実行状態（MOVE）にします。ここまではSUBOUTと同じですが，このあとサブCPUの実行待ち状態（WAIT）になり，実行終了後，サブCPUの設定した中間領域のデータを読み込んでそのバイト数をRCBLNHに書き込み，続いてRCBBMHのバイト数だけデータバッファに書き込みます。

このとき，RCBLNHの値と最初にデータ領域に入っていたデータは消滅しますが，これはデータ領域が仮想中間領域になっているからです。そのためRCBLNHとサブシステムのリクエストデータは，SUBINを使うごとに書き直す必要があります（これはA，Xレジスタでやっています）。SUBINを使うリクエストデータは大体7バイト以下で，長くても30バイト止まりですから，書き直す手間はそれほどかかりません。

もしサブCPUでエラーが生じても（データがしっかりしていれば起こりません！）データ領域のデータはそのままで，RCBSTAにエラー番号が書き込まれません。

以上でRCBの説明は終わりです。

続いてデータ領域のデータの説明をします。

EとCNは，通常0にしておいてかまいません。Cはコマンドコードで，リードタイマのコード$3Eになっています（**図2―5**参照）。RCBLNH，E，CN，Cは，BIOSを呼び出す前に設定しなおしておく必要があります。

復帰情報は，エラーコードE，CN，空文字（00）のあとに各コマンド別に続きます（**図2―6**参照）。ここでは不定値（00），制御レジスタTC，時計レジスタT1，割り

©1984 A. FURUTA

図2―1

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 8E 00 80 BF 40 1D 8E 00 B8 ■._ソ@.■.
4008 00 BF 40 21 86 3E B7 40 DB .ソ@!■>ﾁ@
4010 23 8E 40 19 AD 9F FB FA 4B #■@.ｭﾟ
4018 39 11 00 40 21 00 80 00 2B 9..@!._.
4020 15 00 00 3E 00 00 00 00 53 ...>....
4028 00 00 00 00 00 00 00 00 00 ........
4030 00 00 00 00 00 00 00 00 00 ........
```

図2―2

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 8E 00 80 BF 40 1D 8E 00 B8 ■._ソ@.■.
4008 00 BF 40 21 86 3E B7 40 DB .ソ@!■>ﾁ@
4010 23 8E 40 19 AD 9F FB FA 4B #■@.ｭﾟ
4018 39 11 00 40 21 00 15 00 C0 9..@!...
4020 15 00 00 00 00 08 00 11 2E ........
4028 16 25 00 00 00 00 FF FF 39 .%......
4030 37 77 00 00 00 00 00 00 AE 7w......
```

図2—3

```
01000                                   NAM     SUBIN
01010                                   TTL     ** BIOS SUBIN **
01020   4000                            ORG     $4000
01030                           *
01040                   FBFA    BIOS    EQU     $FBFA
01050                           *
01060                   4000    ENTRY   EQU     *
01070                           *
01080   4000 8E         0080    START   LDX     #128
01090   4003 BF         401D            STX     RCBLNH
01100   4006 8E         0000            LDX     #0
01110   4009 BF         4021            STX     DATA
01120   400C 86         3E              LDA     #$3E
01130   400E B7         4023            STA     C
01140   4011 8E         4019            LDX     #RCB
01150   4014 AD         9F FBFA         JSR     [BIOS]
01160   4018 39                         RTS
01170                           *
01180                   4019    RCB     EQU     *
01190   4019            11      RQNO    FCB     17
01200   401A            00      RCBSTA  FCB     $00
01210   401B            4021    RCBDBA  FDB     DATA
01220   401D            0080    RCBLNH  FDB     128
01230   401F            0015    RCBBMH  FDB     21
01240                           *
01250                   4021    DATA    EQU     *
01260   4021            00      E       FCB     0
01270   4022            00      CN      FCB     0
01280   4023            3E      C       FCB     $3E,$00
        4024            00
01290   4025            00      TC      FCB     $0
01300   4026            00      T1      FCB     $0,$0,$0,$0
        4027            00
        4028            00
        4029            00
01310   402A            00      T1I     FCB     $0,$0,$0,$0
        402B            00
        402C            00
        402D            00
01320   402E            00      T2      FCB     $0,$0,$0,$0
        402F            00
        4030            00
        4031            00
01330   4032            00      T2D     FCB     $0,$0,$0,$0
        4033            00
        4034            00
        4035            00
01340                           *
01350                   4000            END     ENTRY
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=4000
PROGRAM END   ADDR=4035
PROGRAM ENTRY ADDR=4000
```

図2—4

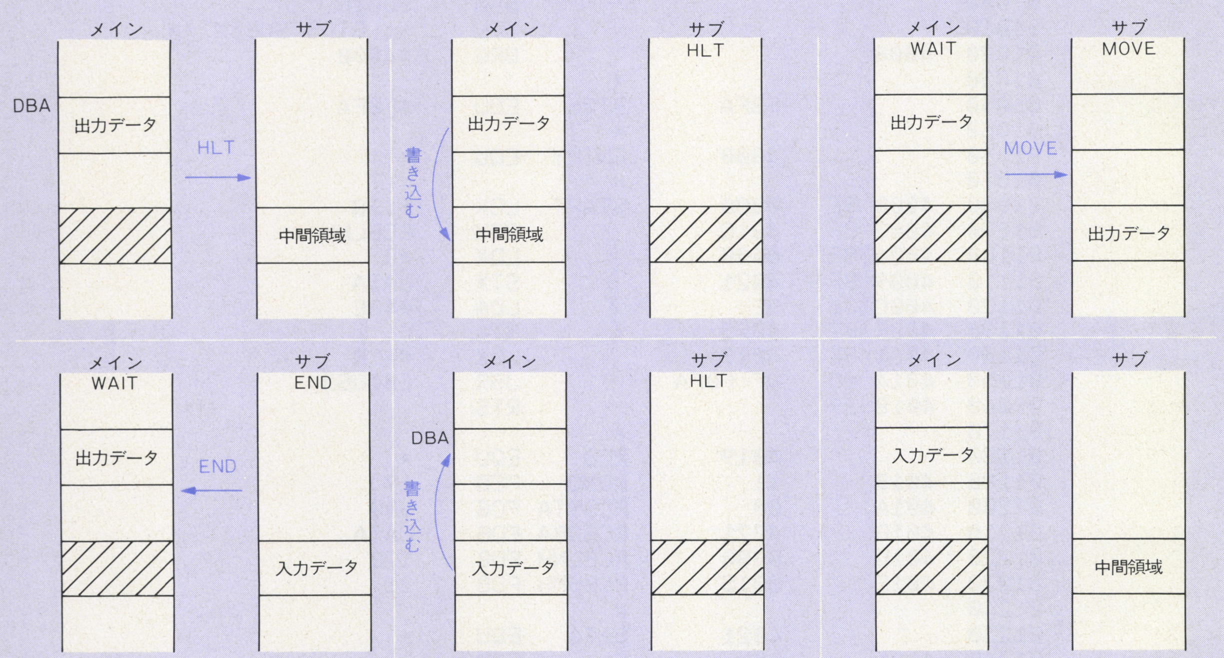


図2—5　コマンド&パラメータ

相対値	記号	内　容	値
0,1	—		Don't care
2	C	コマンドコード	$3E

図2—6　復帰情報

相対値	記　号	内　　容	値
0	E	エラーコード	0
1,2,3	—		不定
4	TC	制御レジスタ	1バイトデータ
5～8	TI	24時間時計レジスタ	4バイトデータ
9～12	TI-I	割り込み予約時刻レジスタ	4バイトデータ
13～16	T2	20msデクリメントカウントレジスタ	32ビットデータ
17～20	T2-D	再設定値レジスタ	32ビットデータ

込み予約時刻レジスタ T1-I, インタバルカウンタ T2, 再設定値レジスタ T2-D がそれです。

復帰情報が128バイトを超えるときは, S=01となるので, SUBIN を何回か使って順次読み込みます。

これも使い方をまとめてみましょう。

①RCB領域にリクエスト番号17($11), エラーステータス領域, データ領域番地, 転送バイト数128, 復帰情報バイト数をセットする。②データ領域に「0, 0, コマンドコード, データ列」を書き込む。③Xレジスタに RCB のアドレスをロードして, BIOS をコールする。④エラーがあると RCBSTA にエラーコードが入り, ないときは復帰情報がデータ領域に書き込まれている。⑤継続情報がある場合は, SUBIN を繰り返し使って, これを読み込む。

なお, ERASE2 とリードタイマ以外のコマンドも, SUBOUT と, SUBIN を使うことで, 自由に使うことができます。

❻ KEYIN編

今度はスピードゲームに欠かせない，リアルタイムキーインです。BASICの「INKEY$」にあたります。

図2—7がダンプリスト，**図2—8**がソースリスト，**図2—9**が実行結果です。

打ち込み終わったら，セーブを忘れないでください。KEYINをコールしただけではKEY情報は空文字になってしまうので，KEYが押されるまで待つようにつくってあります。つまり「INPUT$(1)」と同じ作業をさせています。

＄4000～＄4005は，KEYINのコールとKEY入力判断をしています。KEYINでKEY入力があったときは，B(アキュムレータB）の値が1になるようにBIOS側で設定してあるので，「TSTB／BEQ START」(「IF B＝0 GOTO START」に同じ）を使ってチェックしています。入力がないときはスタートに戻り，あったときはリターンします。

＄4006～＄4012がKEYINコールプログラムです。KEYINのコール後は，データ領域に「入力キーコード」「入力フラグ（0のとき入力なし，1のとき入力あり)」の2バイトのデータが入りますので，これをD(拡張アキュムレータ)で読みとって（**図2—10参照**）リターンします。すなわち，アキュムレータAに入力キーコード，Bに入力フラグが入ります。

ここでKEYINの命令の「PSHS X」「PULS X,PC」について説明しておきます。

PSH命令はレジスタの値を退避させるために使うもので，PSHS Xを実行すると，Xレジスタの値がSスタックの示すアドレスにストアされ，Sスタックの値が2つ減ります(**図2—11参照**)。つまり「LEAS −2，S／STX，S」と同じことをやっています。

次にPULS X,PCを実行すると，Sスタックの示すアドレスから内容がX，PCにロードされ，Sスタックの値が4つ増えます（**図2—12参照**)。つまり「LDX，S／LEAS 2，S／LDPC，S／LEAS 2，S」と同じことをやっています。

なぜこんな面倒なことをするかというと，ロードやストアを使ってレジスタの値をいちいち退避していたのでは，データの

図2—7

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 8D 04 5D 27 FB 39 34 10 8D ■.]´ 94.
4008 8E 40 14 AD 9F FB FA FC 1F ■@.ｭﾉ
4010 40 1A 35 90 15 00 40 1A 8E @.5ﾆ..@.
4018 00 02 00 00 00 00 00 00 02 ........
```

図2—8

```
                                NAM    KEYIN
                                TTL    ** BIOS KEYIN **
        4000                    ORG    $4000
                         *
                FBFA     BIOS   EQU    $FBFA
                         *
                4000     ENTRY  EQU    *
                         *
        4000 8D 04  4006 START  BSR    KEYIN
        4002 5D                 TSTB
        4003 27 FB  4000        BEQ    START
        4005 39                 RTS
                         *
        4006 34 10       KEYIN  PSHS   X
        4008 8E 4014            LDX    #RCB
        400B AD 9F FBFA         JSR    [BIOS]
        400F FC 401A            LDD    DATA
        4012 35 90              PULS   X,PC
                         *
                4014     RCB    EQU    *
        4014    15       RQNO   FCB    21
        4015    00       RCBSTA FCB    $00
        4016    401A     RCBDBA FDB    DATA
        4018    0002     RCBLNH FDB    $0002
        401A    0000     DATA   FDB    $0000
                         *
                4000            END    ENTRY
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=4000
PROGRAM END   ADDR=401B
PROGRAM ENTRY ADDR=4000
```

図2—9

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 8D 04 5D 27 FB 39 34 10 8D ■.]´ 94.
4008 8E 40 14 AD 9F FB FA FC 1F ■@.ｭﾉ
4010 40 1A 35 90 15 00 40 1A 8E @.5ﾆ..@.
4018 00 02 31 01 00 00 00 00 34 ..1.....
```

"1" 入力あり

図2—10 LDD $401A

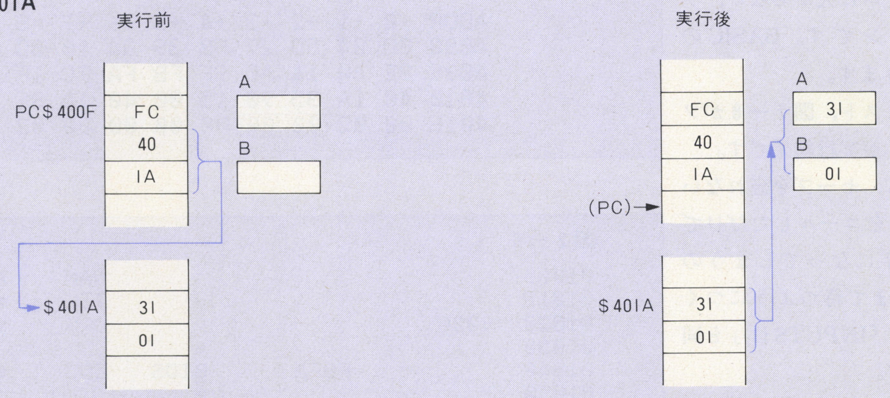


図2—11 PSHS X

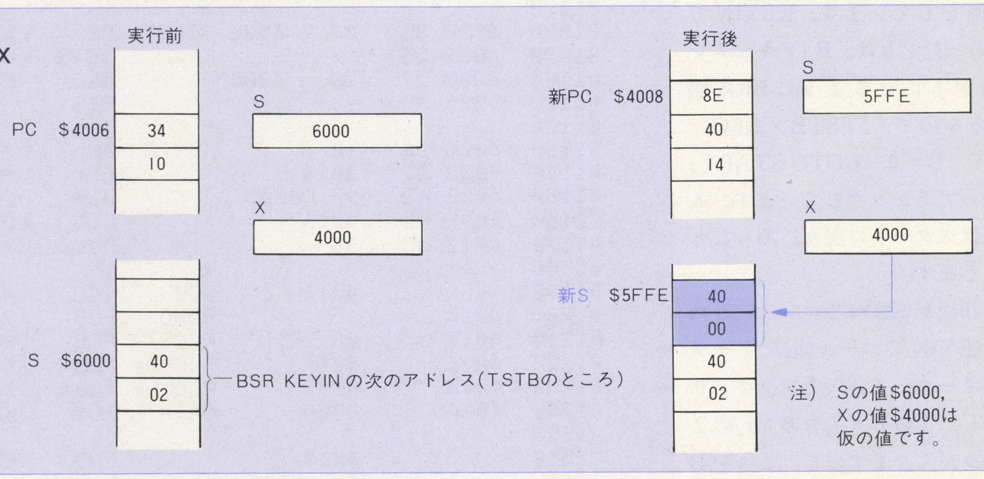


図2—12 PULS X,PC

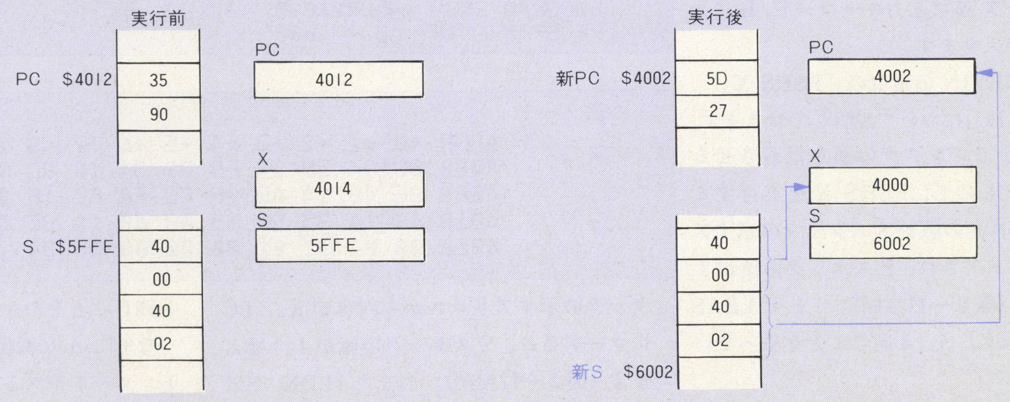


図2―14

```
01000                            NAM    ANALGP
01010                            TTL    ** BIOS ANALGP **
01020  4000                      ORG    $4000
01030                     *
01040          FBFA       BIOS   EQU    $FBFA
01050                     *
01060          4000       ENTRY  EQU    *
01070                     *
01080  4000 34  10        START  PSHS   X
01090  4002 8E  400B             LDX    #RCB
01100  4005 AD  9F FBFA          JSR    [BIOS]
01110  4009 35  90               PULS   X,PC
01120                     *
01130          400B       RCB    EQU    *
01140  400B    00         RQNO   FCB    0
01150  400C    00         RCBSTA FCB    $00
01160  400D    00         RCBCNL FCB    $00
01170  400E    48         RCBRNG FCB    $48
01180  400F    00         RCBAPD FCB    $00
01190                     *
01200          4000              END    ENTRY
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

PROGRAM BEGIN ADDR=4000
PROGRAM END   ADDR=400F
PROGRAM ENTRY ADDR=4000
```

図2―13

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 34 10 8E 40 0B AD 9F FB 64 4.■@.ュ╯
4008 FA 35 90 00 00 00 48 00 07  5ㅗ...H.
```

図2―15

```
ADDR +0 +1 +2 +3 +4 +5 +6 +7 =S 01234567
4000 34 10 8E 40 0B AD 9F FB 64 4.■@.ュ╯
4008 FA 35 90 00 00 00 48 D7 DE  5ㅗ...Hラ
                          入力
```

量が多すぎる場合に，どこにしまったのかわからなくなるからです。

ここでXのPSH，PULをしているのは，KEYINをコールしてもXの値が変わらないようにするためです。

BSRというのは，PC(プログラムカウンタ）を中心として，その前後のサブルーチンをコールする命令です。「8D，04」によって，「PSHS　PC／PC＝PC+#04」という作業を一度にやっています。つまり，PCの値＄4002をSスタックにPSHして，PCに04を加えることでKEYIN（アドレス＄4006)にブランチし，再び「PULS　X，PC」でスタックした値＄4002をロードして「TSTB」へリターンしています。

使い方をまとめます。

①RCBのリクエスト番号21（＄15)，RCBDBAにデータ領域番地，RCBLNHに0002の値をセットする。②データ領域2バイトを用意してBIOSをコールすると，KEY情報がデータ領域に書き込まれる。

⑦ ジョイスティック編

最後に，これもリアルタイム入力に便利なANALGPを紹介しておきましょう。

これは7ではサポートされてませんので，7の人はVol.5の「ジョイスティックの製作」の方を参考にしてください。

図2―13がダンプリスト，**図2―14**がソースリスト，**図2―15**が実行結果です。

リクエストNo. 0，エラーステータス1バイト，アナログチャンネル(RCBCNL) 0～3，電圧レンジ(RCBRNG) H（＄48)またはL（$4C)，データ領域1バイトをセットしてBIOSをコールすると，RCBAPDに入力データが書き込まれます。エラーステータスは，常に正常値（ステックがつながってなくても）を示します。

for FM-7,8,11

効率的なビジネスソフト作成のために

ビジネス用BASICプログラムジェネレータ

〔1〕BASICプログラムを作るプログラムの作り方

高木 征蔵

1. ビジネスソフトの特徴

ビジネスシステムは一般に複雑ですが，この複雑さは対象とする帳票に含まれる各項目の性格による場合が多く，事務全体の流れとしては比較的単純です。請求書の発行の例で考えてみますと，事務の流れは次のようになります。

①売上伝票をまとめて入力する。
②顧客別に並べかえる。
③顧客マスタファイルを索引して，住所，氏名などを取り出す。
④請求書の形でプリントする。

すなわち，データ入力，並べかえ（ソート），マスタ索引，リストという一般的なパターンです。このほかによく使われる事務処理のパターンとして，集計，マスタファイル更新などがあります。

コンピュータによるビジネスシステムが効果を発揮する理由は主に次の2つです。

A．マスタファイルを持つことにより一度入力したデータがいつでも引き出せる。
B．並べかえ，集計，一定の手順の再現が容易にできる。

とくにマスタファイルの作成は重要であり，この設計が終わればそのビジネスシステムは半分完成したようなものです。マスタファイルの書式は千差万別でしょうが，これのメンテナンスは，追加，修正，削除の3つの処理で済み，内容の確認のためのCRTの表示と，ハードコピーとしての印刷を付け加えればすべてOKです。

このように考えてくると，ファイルの書式などケースによって異なるデータを入力するだけでビジネス用BASICプログラムができれば便利だなあという思いにかられます。これを実現しようというのが本稿のねらいです。ここで注意しておきたいのは，簡易言語ではなくBASICのプログラムが作られるという点です。先に述べたようにビジネスシステムは全体としての流れは単純ですが，個々には千差万別です。この千差万別の細かい点をBASICプログラムならば修正によって容易に対処できるからです。特殊な計算式なども苦になりません。

それではビジネスソフトについてのお話はこれくらいにして，BASICでBASICのプログラムを作る方法について考えてみましょう。

2. F-BASICのプログラムファイル

BASICプログラムをフロッピまたはテープにASCII形式でセーブした場合，このプログラムファイルはLINE INPUT #文で読み取ることができます。これとは逆に，PRINT #命令で作成したASCII形式のシーケンシャルファイルは，各々の行の形式がF-BASICの文法に反しなければ，LOADコマンドによりBASICプログラムとして読み込むことができます。

例えばBASICのプログラム“REI-A”が

```
SAVE "REI-A",A
```

によって0番のフロッピにASCII形式でセーブされた後で**図1**のプログラム“BASIC-A”を実行させてみます。実行後コマンドFILESによりディレクトリリストをとると，

```
REI-A  ØAS1
REI-B  1AS1
```

となっていますが，さらに，

```
LOAD "REI-B"
SAVE "REI-B",A
```

を実行した後で再びディレクトリリストをとってみると，

```
REI-A  ØAS1
REI-B  ØAS1
```

となっており，ASCII形式でF-BASICの文法を満足する限り，プログラムファイルかデータファイルかの区別0，1は気にしなくてよいことがわかります。

3. プログラムをプログラムで作る方法

BASICの文法に従った文字列データをASCII形式でシーケンシャルファイルに記入していけばBASICプログラムのファイルが作成されることがわかりました。それではどのような方法でこれを作ったらよいのでしょうか。まず第一に，

(1) 作り出されるプログラムの仕様を定義するデータの入力

が必要です。例えば，作表プログラムならばタイトル，サブタイトル，各行の印刷の書式と変数，入力ファイルのフォーマットなどです。

次に必要となるのは，

(2) 入力されたデータをプログラムに展開するロジック

です。プログラムを作るプログラム（以下プログラムジェネレータと呼ぶ）の主要部分であるロジックの展開方法としては，プログラム自身でロジックを組み立てていく方法と，原形となる標準プログラムを用意してこれを利用する方法が考えられます。ビジネスプログラムの場合，ある程度プログラムのパターン化が可能ですので，基本的には後者を採用するのが妥当でしょう。

したがって，プログラムジェネレータを作成する手順は次のようになります。

①パターンとして汎用性があると判断される原形の標準プログラムを選定する。
②この原形プログラムの中でどの部分が入力されるデータによって置き換えられるべきかを検討する。
③作り出されるプログラムの仕様を定義するデータを決め，これを入力する部分のプログラムを作成する。

©1984 S. TAKAGI

図1　ASCII 形式プログラムファイルの読み取りプログラム

```
10 '    "BASIC-A"
20 OPEN "I",#1,"REI-A"
30 OPEN "O",#2,"REI-B"
40 IF EOF(1) THEN CLOSE:END
50 LINEINPUT #1,A$:PRINT #2,A$:GOTO 40
```

図2　メニュープログラムジェネレータ　MENUGEN

```
10 '    "MENUGEN"    ﾒﾆｭｰ ﾌﾟﾛｸﾞﾗﾑｼﾞｪﾈﾚｰﾀ
20 CLEAR 2000
30 CLS:WIDTH 40,25:COLOR 4:PRINT "   ﾒﾆｭｰ ﾌﾟﾛｸﾞﾗﾑｼﾞｪﾈﾚｰﾀ ":PRINT:COLOR 7
40 INPUT "ｱｳﾄﾌﾟｯﾄ ﾌｧｲﾙ =",O$
50 INPUT "ﾌﾟﾛｸﾞﾗﾑﾒｲ     =",MEI$
60 INPUT "ﾌﾟﾛｸﾞﾗﾑ ｾﾂﾒｲ =",C$
70 INPUT "ﾒﾆｭｰ ﾀｲﾄﾙ     =",MT$
80 INPUT "ﾒﾆｭｰ ﾉ ｶｽﾞ    =",N
90 PRINT " No.  ﾒﾆｭｰ             :ﾌﾟﾛｸﾞﾗﾑ"
100 FOR I=1 TO N
110 PRINT STR$(I)+". ";:LINEINPUT B$:K=INSTR(B$,":"):L=LEN(B$)
120 IF K>1 AND L>3 THEN MENU$(I)=LEFT$(B$,K-1):PRO$(I)=RIGHT$(B$,L-K) ELSE BEEP:
GOTO 110
130 NEXT I
140 INPUT "ﾒﾆｭｰ ﾜｸﾄﾞﾘ ﾓｼﾞ,ｲﾛ =",MOJI$,IRO
150 INPUT "RUN ﾌﾟﾛｸﾞﾗﾑ ｷﾊﾞﾝ  =",RF$
160 '    Program File Output
170 OPEN "O",#1,O$+":"+MEI$
180 PRINT #1,"10 '    "+CHR$(34)+MEI$+CHR$(34)+"   "+C$
190 PRINT #1,"20 CLS:WIDTH 40,25"
200 PRINT #1,"30 COLOR 5:LINE@(2,0)-(38,23),"+CHR$(34)+MOJI$+CHR$(34)+","+STR$(I
RO)+",B"
210 PRINT #1,"40 LOCATE 6,2:COLOR 7:PRINT "+CHR$(34)+MT$+CHR$(34)
220 FOR I=1 TO N
230 PRINT #1,MID$(STR$(10*(I+4)),2,3)+" LOCATE 6,"+STR$(2*I+3)+":PRINT "+CHR$(34
)+STR$(I)+". "+MENU$(I)+CHR$(34)
240 NEXT I
250 PRINT #1,"130 LOCATE 6,21:PRINT "+CHR$(34)+" 9. ｵﾜﾘ"
260 PRINT #1,"140 COLOR 4:LOCATE 10,23:PRINT "+CHR$(34)+"    ﾊﾞﾝｺﾞｳ ｦ ｲﾚﾃｸﾀﾞｻｲ 。
"+CHR$(34)+":COLOR 7"
270 PRINT #1,"150 BG$=INPUT$(1):BG=VAL(BG$)"
280 PRINT #1,"160 IF BG>=1 AND BG<=9 THEN 180"
290 PRINT #1,"170 GOTO 150"
300 FOR I=1 TO N
310 PRINT #1,MID$(STR$(10*(17+I)),2,3)+" IF BG="+STR$(I)+" THEN RUN "+CHR$(34)+R
F$+":"+PRO$(I)+CHR$(34)
320 NEXT I
330 PRINT #1,"270 IF BG= 9 THEN END"
340 PRINT #1,"280 GOTO 150"
350 CLOSE:END
```

④この入力データで原形のプログラムの各行を作り出すロジックを決め，プログラムとして完成させる。

以上が手順の概略ですが，話がだいぶややこしくなってしまいましたので，さらに進んだプログラムジェネレータの作り方の説明はあと回しにして，論より証拠，実際にプログラムジェネレータを動かしてみましょう。

4. メニュープログラムジェネレータ“MENUGEN”

「百聞は一見にしかず」です。まず，図2のプログラムを打ち込んでフロッピまたはテープにセーブしてください。FM-7，8，11のどれでも結構です。これは，個々の業務のメニューを表示し，番号の選択により指定されたプログラムを実行するプログラムのジェネレータで，ビジネスシステムでは通常必ず最初に出てくるものです。

それではメニュープログラムジェネレータ“MENUGEN”の使用法を説明しましょう（図3参照）。

①作成されるメニュー画面の例（図4）にならって，メニュー画面を設計する。メニューは最大8個まで。

②“MENUGEN”をRUNする。

③アウトプット　ファイル　＝　によって，作成されたプログラムを記録する装置をきいてくるので，0，1，CAS0などどれかを指定する。

図3 MENUGENの入力画面の例

```
  ﾒﾆｭｰ ﾌﾟﾛｸﾞﾗﾑｼﾞｪﾈﾚｰﾀ

ｱｳﾄﾌﾟｯﾄ ﾌｧｲﾙ =0
ﾌﾟﾛｸﾞﾗﾑﾒｲ    =ZK-MENU
ﾌﾟﾛｸﾞﾗﾑ ｾﾂﾒｲ =ｾｲﾋﾝ ｻﾞｲｺ ｶﾝﾘ ﾒﾆｭｰ ﾌﾟﾛｸﾞﾗﾑ
ﾒﾆｭｰ ﾀｲﾄﾙ    =◆◆◆ ｾｲﾋﾝ ｻﾞｲｺ ｶﾝﾘ ﾒﾆｭｰ ◆◆◆
ﾒﾆｭｰ ﾉ ｶｽﾞ   =7
 No.  ﾒﾆｭｰ           :ﾌﾟﾛｸﾞﾗﾑ
 1. ｾｲﾋﾝ ﾏｽﾀｰ ﾒﾝﾃﾅﾝｽ:SM-MENTE
 2. ｳｹｲﾚ, ﾊﾗｲﾀﾞｼ ﾃﾞｰﾀ ｲﾝﾌﾟｯﾄ:DATAIN
 3. ｲﾝﾌﾟｯﾄ ﾃﾞｰﾀ ﾁｪｯｸ ﾘｽﾄ:CHECKHYO
 4. ｻﾞｲｺ ｺｳｼﾝ ｼｮﾘ:KOSIN
 5. ｻﾞｲｺﾋｮｳ ﾘｽﾄ:ZAIKOHYO
 6. ｻﾞｲｺ ﾄｲｱﾜｾ:TOIAWASE
 7. ｹﾞｯｶﾝ ｼｭｳｹｲﾋｮｳ ﾘｽﾄ:GEPPO
ﾒﾆｭｰ ﾜｸﾄﾞﾘ ﾓｼﾞ,ｲﾛ =*,4
RUN ﾌﾟﾛｸﾞﾗﾑ ｷﾊﾞﾝ  =0
```

④指定した装置に媒体をセットし，READY?に対しててYを入力する。

⑤プログラムメイ　=　に対し，作り出されるメニュープログラム名を入力する。

⑥プログラム　セツメイ　=　に対し，説明文を入力する。

⑦メニュー　タイトル　=　に対し，①で設計したメニュー画面のタイトルを入力する。

⑧メニュー　ノ　カズ　=　に対し，①で設計したメニューの数（1～8の間の数字）を入力する（9．オワリは自動挿入される）。

⑨①で設計したメニューの各番号の文章とその番号を選んだときにRUNするプログラムの名前を「：」で区切って入力する。

⑩メニュー　ワクドリ　モジ，イロ　=　に対し，枠取りの文字と色コードを「，」で区切って入力する（枠取りしない場合は例えば␣，7と入力。ここで␣はスペースの意味）。

⑪RUN　プログラム　キバン　=　に対し，メニュー番号を選択したときにRUNするプログラムを格納しておく装置の機番(CAS0, 0, 1など)を入力する。

⑫メニュープログラムが作成され，④で指定した媒体に記録される。

図4 作成されるメニューの画面の例

```
*************************************
*                                   *
*   ◆◆◆ ｾｲﾋﾝ ｻﾞｲｺ ｶﾝﾘ ﾒﾆｭｰ ◆◆◆     *
*                                   *
*                                   *
*    1. ｾｲﾋﾝ ﾏｽﾀｰ ﾒﾝﾃﾅﾝｽ             *
*                                   *
*    2. ｳｹｲﾚ, ﾊﾗｲﾀﾞｼ ﾃﾞｰﾀ ｲﾝﾌﾟｯﾄ     *
*                                   *
*    3. ｲﾝﾌﾟｯﾄ ﾃﾞｰﾀ ﾁｪｯｸ ﾘｽﾄ         *
*                                   *
*    4. ｻﾞｲｺ ｺｳｼﾝ ｼｮﾘ                *
*                                   *
*    5. ｻﾞｲｺﾋｮｳ ﾘｽﾄ                  *
*                                   *
*    6. ｻﾞｲｺ ﾄｲｱﾜｾ                   *
*                                   *
*    7. ｹﾞｯｶﾝ ｼｭｳｹｲﾋｮｳ ﾘｽﾄ           *
*                                   *
*                                   *
*                                   *
*    9. ｵﾜﾘ                          *
*                                   *
********    ﾊﾞﾝｺﾞｳ ｦ ｲﾚﾃｸﾀﾞｻｲ 。******
```

図5 図3の入力で作成されたプログラム ZK-MENU

```
10 '    "ZK-MENU"  ｾｲﾋﾝ ｻﾞｲｺ ｶﾝﾘ ﾒﾆｭｰ ﾌﾟﾛｸﾞﾗﾑ
20 CLS:WIDTH 40,25
30 COLOR 5:LINE@(2,0)-(38,23),"*", 4,B
40 LOCATE 6,2:COLOR 7:PRINT "◆◆◆ ｾｲﾋﾝ ｻﾞｲｺ ｶﾝﾘ ﾒﾆｭｰ ◆◆◆"
50 LOCATE 6, 5:PRINT " 1. ｾｲﾋﾝ ﾏｽﾀｰ ﾒﾝﾃﾅﾝｽ"
60 LOCATE 6, 7:PRINT " 2. ｳｹｲﾚ, ﾊﾗｲﾀﾞｼ ﾃﾞｰﾀ ｲﾝﾌﾟｯﾄ"
70 LOCATE 6, 9:PRINT " 3. ｲﾝﾌﾟｯﾄ ﾃﾞｰﾀ ﾁｪｯｸ ﾘｽﾄ"
80 LOCATE 6, 11:PRINT " 4. ｻﾞｲｺ ｺｳｼﾝ ｼｮﾘ"
90 LOCATE 6, 13:PRINT " 5. ｻﾞｲｺﾋｮｳ ﾘｽﾄ"
100 LOCATE 6, 15:PRINT " 6. ｻﾞｲｺ ﾄｲｱﾜｾ"
110 LOCATE 6, 17:PRINT " 7. ｹﾞｯｶﾝ ｼｭｳｹｲﾋｮｳ ﾘｽﾄ"
130 LOCATE 6,21:PRINT " 9. ｵﾜﾘ
140 COLOR 4:LOCATE 10,23:PRINT "    ﾊﾞﾝｺﾞｳ ｦ ｲﾚﾃｸﾀﾞｻｲ 。":COLOR 7
150 BG$=INPUT$(1):BG=VAL(BG$)
160 IF BG>=1 AND BG<=9 THEN 180
170 GOTO 150
180 IF BG= 1 THEN RUN "0:SM-MENTE"
190 IF BG= 2 THEN RUN "0:DATAIN"
200 IF BG= 3 THEN RUN "0:CHECKHYO"
210 IF BG= 4 THEN RUN "0:KOSIN"
220 IF BG= 5 THEN RUN "0:ZAIKOHYO"
230 IF BG= 6 THEN RUN "0:TOIAWASE"
240 IF BG= 7 THEN RUN "0:GEPPO"
270 IF BG= 9 THEN END
280 GOTO 150
```

図6　例題1　リストフォーム

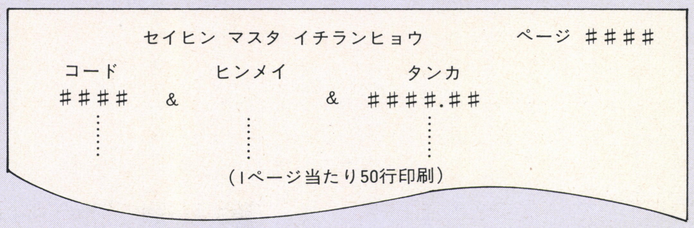
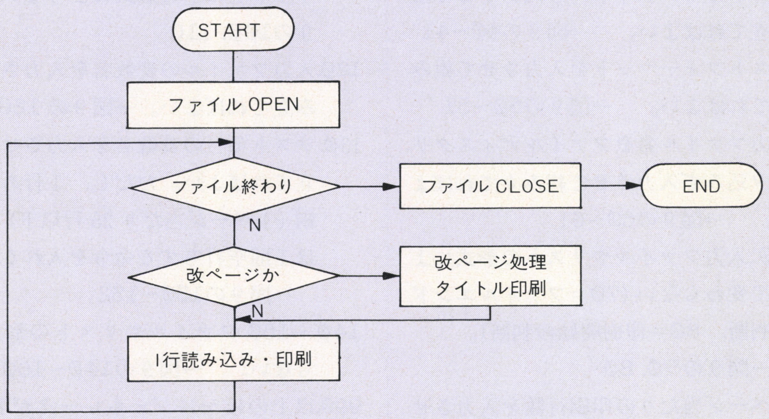


図7　例題1　フローチャート



これでメニュープログラムが作成されました。FILES コマンドで確認してみてください。ASCII 形式のシーケンシャルデータファイルとして記録されていますが，

RUN "装置名：プログラム名"

で走らせてください。あなたの設計したメニュー画面が表示されるでしょう。

データの入力ミス，あるいはメニュー画面の修正は作成されたプログラムを普通の方法で修正してください。修正後セーブするときに Y を入力することでジェネレータの作成したオリジナルのプログラムファイル（ディレクトリではデータファイル）は消え，新しいファイルがプログラムファイルとして登録されます。

実行例として，**図3**の入力で作成されたプログラム ZK-MENU を**図5**に示します。これを実行させると**図4**のメニューが表示されます。

5. プログラムジェネレータの作成例

図2のプログラム"MENUGEN"とあなたが"MENUGEN"で作成したプログラムとを対比されると，プログラムジェネレータの作り方は大体理解されると思いますが，ここで次のような問題から出発して実際にジェネレータを作ってみましょう。

例題1．　製品マスタのリスト

製品マスタファイル

- シーケンシャルファイル
- ファイル名　SEIHIN-M
- フロッピ　0番にセット
- 品名コード，品名，単価が1レコードに記入されている

リストフォームは**図6**のとおり

この例題を**図7**のフローチャートに従ってプログラムにすると**図8**のようになります。この種のプログラムはビジネス用途としては無数に作られますが，一般にはその都度どこを変えたらよいか考えつつ，必要部分を追加・修正・削除して完成させる方法がとられます。そこでこの判断をプログラムによって行い，ファイルの変数，リストの形式などのデータを入力するだけでプログラムが作成されるようにしようというのが，プログラムジェネレータのやり方です。それでは**図8**のプログラムを基本にしてジェネレータを作成してみましょう。作成されたプログラムはフロッピ0番に記録するものとします。

プログラム"REIDAI-1"の行番号を1つずつ追って検討してみましょう（文の前の番号は**図8**の行番号です）。

10. プログラム名，プログラムの説明を入力させて組み立てればよい。入力されたプログラム名を出力ファイル名として使えばよい。　→**図9**の10～13

20. 実行中の作業を CRT に表示するだけであり，プログラムの説明をそのまま利用すればよい。　→図9の20

30. リストのヘッドタイトルを入力させて

図8　例題1のプログラム

```
10 '    "REIDAI-1"    ｾｲﾋﾝ ﾏｽﾀｰ ﾘｽﾄ
20 WIDTH 40,25:PRINT "ｾｲﾋﾝ ﾏｽﾀｰ ﾘｽﾄ"
30 HD$="          ｾｲﾋﾝ ﾏｽﾀｰ ｲﾁﾗﾝﾋｮｳ                  ﾍﾟｰｼﾞ ####"
40 SHA$=" ｺｰﾄﾞ      ﾋﾝﾒｲ        ﾀﾝｶ"
50 MS$=" ####  &          & ####.##"
60 OPEN "I",#1,"SEIHIN-M":I=1
70 IF EOF(1) THEN 150
80 IF I=1 THEN 100
90 IF I MOD 50 =1 THEN LPRINT CHR$(12):GOTO 100 ELSE 120
100 LPRINT:LPRINT:LPRINT:PG=PG+1:LPRINT USING HD$;PG
110 LPRINT:LPRINT:LPRINT SHA$:LPRINT
120 INPUT #1;SCODE,HINMEI$,NEDAN
130 LPRINT USING MS$;SCODE,HINMEI$,NEDAN
140 I=I+1:GOTO 70
150 CLOSE #1
160 END
```

▒▒は図9のジェネレータにしたとき，置き換えられるもの

組み立てればよい。 →図9の3Ø～32

4Ø. リストのサブタイトルを入力させて組み立てればよい。 →図9の4Ø～41

5Ø. リストフォーマットを入力させて組み立てればよい。 →図9の5Ø～51

6Ø. 入力ファイル名をファイルディスクリプタ込みで入力させて組み立てればよい。 →図9の6Ø～61

7Ø, 8Ø. 入力ファイルやリストの形式によって変わらない (7Ø＝ファイルエンドの判断, 8Ø＝印刷開始の判断)。
→図9の7Ø, 8Ø

9Ø. 1ページ当たりの印刷行数を入力させて組み立てればよい。→図9の9Ø～91

1ØØ, 11Ø. 改ページ処理であり, ファイルやリストの形式によらない。→図9の1ØØ, 11Ø

12Ø. 入力ファイルの変数名を入力させて組み立てればよい。→図9の12Ø～121

13Ø. リスト項目の変数名を入力させて組み立てればよい。ただし, 1行あきで印刷 (1ページ当たり25行以下) の場合は1回空打ちする命令を入れる。
→図9の13Ø～132

14Ø～16Ø. ファイルやリストの形式によらない。 →図9の14Ø～16Ø

999. 以上のほかにジェネレータがBASICのプログラムとして形をなすのに必要な命令を追加する。
前部→1～3 後部5ØØ

これで明細リストプログラムのジェネレータ"MLISTGEN"ができあがりました(**図9**)。形を整えたい方はデータ入力の部分を前半に, PRINT #文を後半にまとめてRENUMしてください。**図2**のメニュープログラムジェネレータと同じスタイルに納まります。

それでは**図8**のプログラムをまず打ち込んでセーブし, それを修正する形で**図9**のプログラムを作成してください。プログラムジェネレータ"MLISTGEN"の使用法は説明しなくとも分かると思います。実際に

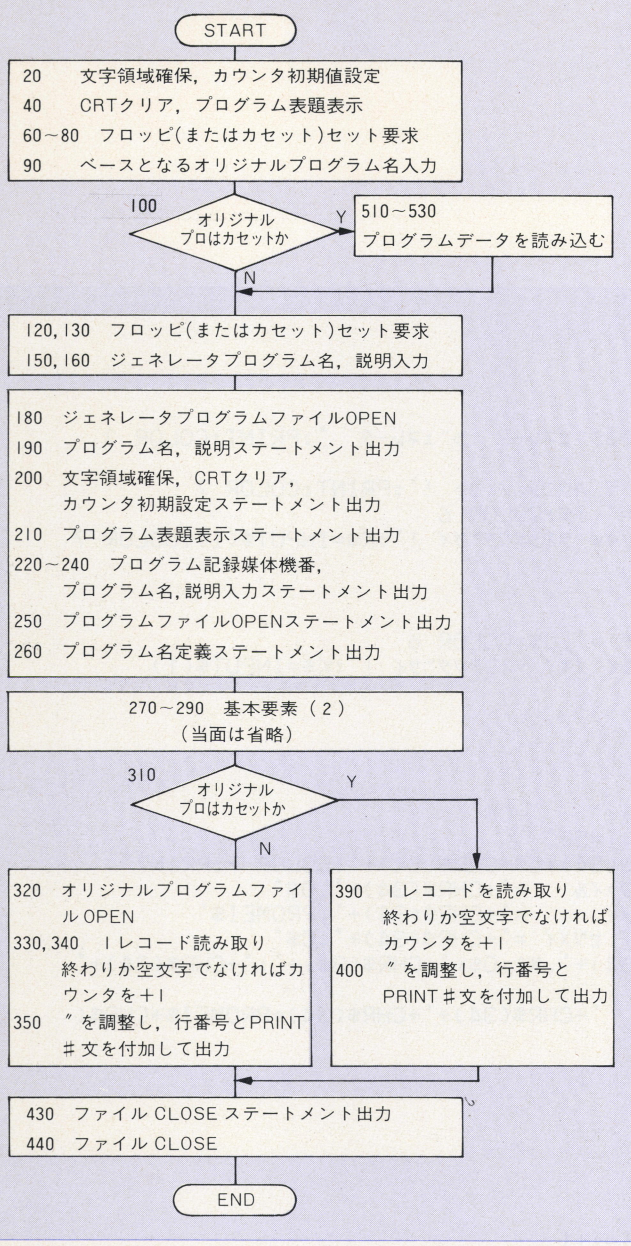
図9 プログラムジェネレータの作成例

```
1 '    "MLISTGEN"   ﾒｲｻｲ ﾘｽﾄ ﾌﾟﾛｸﾞﾗﾑ ｼﾞｪﾈﾚｰﾀ  ﾉ ﾚｲ
2 CLEAR 5000:CLS:WIDTH 80,25
3 COLOR 4:PRINT TAB(5),"ﾒｲｻｲ ﾘｽﾄ ﾌﾟﾛｸﾞﾗﾑ ｼﾞｪﾈﾚｰﾀ ":COLOR 7:PRINT
10 INPUT "ﾌﾟﾛｸﾞﾗﾑ ﾒｲ      =";PROMEI$
11 INPUT "ﾌﾟﾛｸﾞﾗﾑ ﾉ ｾﾂﾒｲ =";C$
12 OPEN "O",#1,PROMEI$
13 PRINT #1,"10 '    "+CHR$(34)+PROMEI$+CHR$(34)+"    "+C$
20 PRINT #1,"20 WIDTH 40,25:PRINT "+CHR$(34)+C$+CHR$(34)
30 LINEINPUT "ﾍｯﾄﾞ ﾀｲﾄﾙ        =";HD$
31 HD$=HD$+"                    ﾍﾟｰｼﾞ ####"
32 PRINT #1,"30 HD$="+CHR$(34)+HD$+CHR$(34)
40 LINEINPUT "ｻﾌﾞ  ﾀｲﾄﾙ        =";SHA$
41 PRINT #1,"40 SHA$="+CHR$(34)+SHA$+CHR$(34)
50 LINEINPUT "ﾒｲｻｲ ﾘｽﾄ  ﾌｫｰﾏｯﾄ  =";MS$
51 PRINT #1,"50 MS$="+CHR$(34)+MS$+CHR$(34)
60 INPUT "ｲﾝﾌﾟｯﾄ ﾌｧｲﾙ ﾒｲ =";FMEI$
61 PRINT #1,"60 OPEN "+CHR$(34)+"I"+CHR$(34)+",#1,"+CHR$(34)+FMEI$+CHR$(34)+":I=
1"
70 PRINT #1,"70 IF EOF(1) THEN 150"
80 PRINT #1,"80 IF I=1 THEN 100"
90 INPUT "1 ﾍﾟｰｼﾞ ﾉ ｷﾞｮｳｽｳ  =";GYOSU
91 PRINT #1,"90 IF I MOD"+STR$(GYOSU)+" =1 THEN LPRINT CHR$(12):GOTO 100 ELSE 12
0"
100 PRINT #1,"100 LPRINT:LPRINT:LPRINT:PG=PG+1:LPRINT USING HD$;PG"
110 PRINT #1,"110 LPRINT:LPRINT:LPRINT SHA$:LPRINT"
120 LINEINPUT "ｲﾝﾌﾟｯﾄ ﾌｧｲﾙ ﾍﾝｽｳ ﾒｲ  =";FMAT$
121 PRINT #1,"120 INPUT #1;"+FMAT$
130 LINEINPUT "ﾒｲｻｲ ﾘｽﾄ  ﾍﾝｽｳﾒｲ  =";HENSU$
131 IF GYOSU>25 THEN Q$="" ELSE Q$=":LPRINT"
132 PRINT #1,"130 LPRINT USING MS$;"+HENSU$+Q$
140 PRINT #1,"140 I=I+1:GOTO 70"
150 PRINT #1,"150 CLOSE #1"
160 PRINT #1,"160 END"
500 CLOSE:END
```

図10　プログラムジェネレータの原形を作るプログラム

```
10 ′    "GENEGENE"    ﾌﾟﾛｸﾞﾗﾑｼﾞｪﾈﾚｰﾀ  ｼﾞｪﾈﾚｰﾀ
20 CLEAR 15000:I=51
30 DIM R$(200)
40 CLS:WIDTH 80,25:COLOR 4:PRINT "     ﾌﾟﾛｸﾞﾗﾑｼﾞｪﾈﾚｰﾀ  ｼﾞｪﾈﾚｰﾀ  ":PRINT:COLOR 2
50 ′
60 PRINT "ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾉ ﾌｧｲﾙ  ASCII SAVE ｦ ｶｸﾆﾝ ﾉ ｺﾄ !":PRINT:COLOR 7
70 INPUT "ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ｦ ｾｯﾄ ｽﾙ ｿｳﾁ =",I$:COLOR 6
80 PRINT "ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ｦ ｾｯﾄ ｼ, ﾆﾝｲ ﾉｷｲ ｦｵｼﾃｸﾀﾞｻｲ !":K$=INPUT$(1):COLOR 7
90 PRINT:INPUT "ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑﾒｲ    =",SP$
100 IF I$="CAS0" THEN GOSUB 510
110 ′
120 PRINT:INPUT "ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑ ｦ ｶｷｺﾑ ｿｳﾁ =",O$:COLOR 6
130 PRINT "ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ｦ ｾｯﾄ ｼ, ﾆﾝｲ ﾉｷｲ ｦｵｼﾃｸﾀﾞｻｲ !":K$=INPUT$(1)
140 COLOR 7:PRINT
150 INPUT "ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑﾒｲ     =",GP$
160 INPUT "ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑ ｾﾂﾒｲ =",C$
170 ′    ｷﾎﾝ ﾖｳｿ (3)   (ｻｲｼｮ ﾉ ﾌﾞﾌﾞﾝ )
180 OPEN "O",#2,O$+":"+GP$
190 PRINT #2,"10 ′    "+CHR$(34)+GP$+CHR$(34)+"    "+C$
200 PRINT #2,"20 CLEAR 5000:CLS:WIDTH 80,25:I=50"
210 PRINT #2,"30 COLOR 4:PRINT TAB(5),"+CHR$(34)+C$+CHR$(34)+":COLOR 7:PRINT"
220 PRINT #2,"40 INPUT "+CHR$(34)+"ｱｳﾄﾌﾟｯﾄ ﾌｧｲﾙ    ="+CHR$(34)+",O$"
230 PRINT #2,"50 INPUT "+CHR$(34)+"ﾌﾟﾛｸﾞﾗﾑ ﾒｲ       ="+CHR$(34)+",PROMEI$"
240 PRINT #2,"60 INPUT "+CHR$(34)+"ﾌﾟﾛｸﾞﾗﾑ ﾉ ｾﾂﾒｲ ="+CHR$(34)+",C$"
250 PRINT #2,"500 OPEN "+CHR$(34)+"O"+CHR$(34)+",#1,O$+"+CHR$(34)+":"+CHR$(34)+"
+PROMEI$"
260 PRINT #2,"510 PRINT #1,"+CHR$(34)+"1 ′    "+CHR$(34)+"+CHR$(34)+PROMEI$+CHR$(
34)+"+CHR$(34)+"    "+CHR$(34)+"+C$"
270 ′    ｷﾎﾝ ﾖｳｿ (2)
280 ′
290 ′
300 ′    ｷﾎﾝ ﾖｳｿ (1)   (ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾉ ｼｮﾘ)
310 IF I$="CAS0" THEN 380
320 OPEN "I",#1,I$+":"+SP$
330 IF EOF(1) THEN 420
340 LINEINPUT #1,A$:IF A$="" THEN 330 ELSE I=I+1
350 GOSUB 450:PRINT #2,MID$(STR$(10*I),2,5)+" PRINT #1,"+CHR$(34)+A$+CHR$(34)
360 GOTO 330
370 ′
380 FOR S=1 TO NSTEP
390 A$=R$(S):IF A$="" THEN 410 ELSE I=I+1
400 GOSUB 450:PRINT #2,MID$(STR$(10*I),2,5)+" PRINT #1,"+CHR$(34)+A$+CHR$(34)
410 NEXT S
420 ′    ｷﾎﾝ ﾖｳｿ (3)   (ｵﾜﾘ ﾉ ﾌﾞﾌﾞﾝ )
430 I=I+1:PRINT #2,MID$(STR$(10*I),2,5)+" CLOSE:END"
440 CLOSE:CLEAR 300:END
450 ′    "  " ﾉ ﾁｮｳｾｲ
460 B$="":L=LEN(A$):K=1:M=INSTR(K,A$,CHR$(34)):IF M=0 THEN RETURN ELSE 480
470 M=INSTR(K,A$,CHR$(34)):IF M=0 THEN 500
480 B$=B$+MID$(A$,K,M-K)+CHR$(34)+"+CHR$(34)+"+CHR$(34)
490 K=M+1:IF K>L THEN A$=B$:RETURN ELSE GOTO 470
500 B$=B$+MID$(A$,K,L-K+1):A$=B$:RETURN
510 ′    ｶｾｯﾄ ﾃﾞｰﾀ ﾖﾐｺﾐ Sub
520 OPEN "I",#1,I$+":"+SP$
530 IF EOF(1) THEN CLOSE #1:RETURN ELSE NSTEP=NSTEP+1:LINEINPUT #1,R$(NSTEP):GOT
O 530
```

図11 GENEGENEのフローチャート



テストして REIDAI-1が作り出されることを確認してください。フロッピとプリンタをお持ちでない方は入力ファイルがカセット，出力ファイルをディスプレイに変えてテストしてください。

6. プログラムジェネレータ作成のための技法

実際に“MLISTGEN”をパンチされた方はお気付きのように，ジェネレータプログラムの基本要素は次の３つです。

(1) 原形となるプログラムのステートメントを“　”で囲み PRINT＃する。

(2) ファイルやリストの形式など，個々の場合で異なるデータの入力要求文を付加する。

(3) プログラムジェネレータとして共通のステートメントを付加する。

前節の例題は短かったので，１つ１つPRINT＃文と“　”を挿入する手間はさほどではありませんが，ベースとなるプログラムが長い場合はたいへんです。また，もとのステートメントの中に「“」がある場合はCHR$(34)を使って処理し，全体としてPRINT＃文が正しくなるようにしなければなりませんが，これはかなりめんどうな作業です。そこで，ベースとなるプログラムをASCII形式でセーブしておき，これをLINE INPUT＃文で１行読み込み，前に行番号と「PRINT#1,“」を，後ろに「”」を付加し，さらにPRINT#文が全体として正しくなるように「“」とCHR$(34)とを適宜挿入し，ASCII形式のシーケンシャルファイルとして書き出せば，プログラムジェネレータの基本要素(1)の原形を作成することができます。これによってキーボードをたたく手間と「“」の処理に頭を使う労力が大幅に軽減されるでしょう。

次に基本要素(2)について考えてみましょう。これは個々のプログラムパターンによって異なるのが原則であり一般化することは困難ですが，ファイルの形式，リストのフォーマット，データのインプットなどビジネスソフトとして出現頻度の高いものについては前もって織り込んでおくと，これらを使用するような場合には後の手仕上げが楽になります。

基本要素(3)はどのジェネレータ にも共通なステートメントで，手紙でいえば〝前略〃，〝草々〃に相当するもので，ジェネレータ作成の都度パンチあるいはコピーするのもおもしろくありません。

以上の考え方をまとめますと，ASCII形式でセーブされた原形となるプログラムのファイルを読み込んで，プログラムジェネレータの粗い形のものを作成するプログラムを作ることができます。プログラムジェネレータの原形を作るジェネレータとして，このプログラムを“GENEGENE”と名付けることにしましょう。詳しい説明は抜きにして“GENEGENE”のソースコードを**図10**に，フローチャートを**図11**に示します。ただし，基本要素(2)の部分は当面除いてあります。この部分についてはこれからの連載の途中で整理される都度，**図10**のプログラムに追加する方法を説明します。フローチャートにソースコードの行番号が記入してあるので，これを対応させてプログラムを解読してみてください。ビジネスソフトを作るジェネレータのジェネレータであることをよく頭に入れてフォローしてください。最初は頭が多少混乱するでしょうが，ジェネレータの概念に慣れるにつれて理解が進むと思います。

それではプログラムジェネレータの原形を作るプログラム“GENEGENE”の使用法を説明しましょう。FM-7，8，11のいずれでも構いません。**図10**のソースコードをパンチしてセーブしてください。外部記憶装置としてカセットレコーダしかない場合は，オリジナルプログラムを一度全部メモ

図12 GENEGENEの入力画面の例

```
      ﾌﾟﾛｸﾞﾗﾑｼﾞｪﾈﾚｰﾀ  ｼﾞｪﾈﾚｰﾀ

ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾉ ﾌｧｲﾙ  ASCII SAVE ｦ ｶｸﾆﾝ ﾉ ｺﾄ !

ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ｦ ｾｯﾄ ｽﾙ ｿｳﾁ =0
ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ｦ ｾｯﾄ ｼ, ﾆﾝｲ ﾉｷｲ ｦｵｼﾃｸﾀﾞｻｲ !

ｵﾘｼﾞﾅﾙ ﾌﾟﾛｸﾞﾗﾑﾒｲ     =REIDAI-1

ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑ ｦ ｶｷｺﾑ ｿｳﾁ =0
ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ｦ ｾｯﾄ ｼ, ﾆﾝｲ ﾉｷｲ ｦｵｼﾃｸﾀﾞｻｲ !

ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑﾒｲ     =REIGEN
ｼﾞｪﾈﾚｰﾀ  ﾌﾟﾛｸﾞﾗﾑ ｾﾂﾒｲ =ﾒｲｻｲ ﾘｽﾄ ﾌﾟﾛｸﾞﾗﾑ ｼﾞｪﾈﾚｰﾀ
```

リに読み込んでから処理するため，あまり大きいプログラムのジェネレータは作れません。外部記憶装置はできれば2台欲しいものです。

①オリジナルプログラムがASCII形式でセーブされていることを確認する（ここではオリジナルプログラムとして図8の“REIDAI-1”を使用する）。

②オリジナルプログラムの記録されている媒体をセットする装置(CAS 0, 0, 1など）を入力する。

③オリジナルプログラムの記録されている媒体をセットし任意のキーを押す。

④オリジナルプログラム名を入力する（カセットレコーダのみの場合はここでオリジナルプログラムをメモリにすべて読み込むためしばらく待つ)。

⑤作り出されるジェネレータプログラムを書き込む装置を入力する。

⑥ジェネレータプログラムを記録する媒体をセットし，任意のキーを押す。

⑦ジェネレータプログラム名を入力する。

⑧ジェネレータプログラムの説明文を入力する。

⑨ジェネレータプログラムが作成されて終わる。

入力例を図12に示します。ジェネレータプログラムの原形が作成されたことを，FILESコマンドおよび実際にプログラムをリストして確認してください。図12の入力の結果作成されたプログラムを図13に示します。図9のプログラムと比較してください。図13を手直しすれば容易に図9のプログラムが得られることがわかるでしょう。

7. これからの予定

だいぶ急ぎましたが，今回の説明でプログラムジェネレータの概要を理解していただけたと思います。次回以降は，本稿で説明した手法によって筆者が製作したプログラムジェネレータをいくつか紹介し，それを用いてビジネスシステムを作成する方法を解説します。今回説明したメニュープログラムジェネレータはその第一歩です。データ入力プロ，各種作表プロ，集計および並べかえ，マスタファイル更新プロなどビジネス処理の基本要素について連載していく予定です。お楽しみに！

図13 図9をオリジナルプログラムとしてGENEGENEで作成したプログラムジェネレータの原形

```
10 '    "REIGEN"   ﾒｲｻｲ ﾘｽﾄ ﾌﾟﾛｸﾞﾗﾑ ｼﾞｪﾈﾚｰﾀ
20 CLEAR 5000:CLS:WIDTH 80,25:I=50
30 COLOR 4:PRINT TAB(5),"ﾒｲｻｲ ﾘｽﾄ ﾌﾟﾛｸﾞﾗﾑ ｼﾞｪﾈﾚｰﾀ ":COLOR 7:PRINT
40 INPUT "ｱｳﾄﾌﾟｯﾄ ﾌｧｲﾙ    =",O$
50 INPUT "ﾌﾟﾛｸﾞﾗﾑ ﾒｲ      =",PROMEI$
60 INPUT "ﾌﾟﾛｸﾞﾗﾑ ﾉ ｾﾂﾒｲ =",C$
500 OPEN "O",#1,O$+":"+PROMEI$
510 PRINT #1,"1 '    "+CHR$(34)+PROMEI$+CHR$(34)+"    "+C$
520 PRINT #1,"10 '    "+CHR$(34)+"REIDAI-1"+CHR$(34)+"    ｾｲﾋﾝ ﾏｽﾀｰ ﾘｽﾄ"
530 PRINT #1,"20 WIDTH 40,25:PRINT "+CHR$(34)+"ｾｲﾋﾝ ﾏｽﾀｰ ﾘｽﾄ"+CHR$(34)+""
540 PRINT #1,"30 HD$="+CHR$(34)+"             ｾｲﾋﾝ ﾏｽﾀｰ ｲﾁﾗﾝﾋｮｳ                 ﾍﾟｰｼﾞ ###
#"+CHR$(34)+""
550 PRINT #1,"40 SHA$="+CHR$(34)+" ｺｰﾄﾞ    ﾋﾝﾒｲ          ﾀﾝｶ"+CHR$(34)+""
560 PRINT #1,"50 MS$="+CHR$(34)+" ####  &         & ####.##"+CHR$(34)+""
570 PRINT #1,"60 OPEN "+CHR$(34)+"I"+CHR$(34)+",#1,"+CHR$(34)+"SEIHIN-M"+CHR$(34
)+":I=1"
580 PRINT #1,"70 IF EOF(1) THEN 150"
590 PRINT #1,"80 IF I=1 THEN 100"
600 PRINT #1,"90 IF I MOD 50 =1 THEN LPRINT CHR$(12):GOTO 100 ELSE 120"
610 PRINT #1,"100 LPRINT:LPRINT:LPRINT:PG=PG+1:LPRINT USING HD$;PG"
620 PRINT #1,"110 LPRINT:LPRINT:LPRINT SHA$:LPRINT"
630 PRINT #1,"120 INPUT #1;SCODE,HINMEI$,NEDAN"
640 PRINT #1,"130 LPRINT USING MS$;SCODE,HINMEI$,NEDAN"
650 PRINT #1,"140 I=I+1:GOTO 70"
660 PRINT #1,"150 CLOSE #1"
670 PRINT #1,"160 END"
680 CLOSE:END
```

FM-7を解析する

F-BASIC V3.0のデータ構造とアルゴリズム

Q-Labolatory

●連載をはじめるにあたって●

今号から何回かにわたってFM-7に搭載されているF-BASIC Version3.0のROM解析を連載することになりますが，今回の解析の対象はF-BASICシステムに限るとし，サブシステムについては除外することを最初にお断りしておきます。

FM-7は，CRTディスプレイやキーボードなどの入出力装置の管理，カセットやディスクなどの補助記憶装置の管理，各種のデータやプログラムの管理を，それぞれメインシステムおよびサブシステムと呼ばれる2つのシステムに分担させ処理させています。なかでも，サブシステムには，人がコンピュータシステムと対話する上で最も重要なCRTディスプレイ，およびキーボードに関するハードウェア資源とソフトウェア資源の管理がゆだねられており，この解析を行わないのは片手落ちとの感をまぬがれません。とくに，高速性が必要なビデオゲームを作成する場合には，サブシステムの解析は避けて通れないことを考えると，一層その感を深くします。しかし，サブシステムについては，雑誌などでソースリストが公開されていたり，その解説も多いようなので，今回はとりあえず，読者のみなさんの手にゆだねることにしました。この点，ご了解願いたいと思います。

今回，解析の対象としたF-BASICシステムは，ディスクBASICを含めて約32Kバイトの大きさのプログラムです。このプログラムには，プログラムの暗号化（Pオプションを指定したSAVEコマンドの実行時），スプーリングによる多重プログラミング(PLAY文実行)，チェインによるデータ構造(RENUMコマンド実行時に作成されるリファレンステーブル）など，興味ある技法が数多くみられます。また，乱数の生成，数値関数の演算，浮動小数点や多項式の演算，FOR～NEXT文やWHILE～WEND文の制御などを実現するアルゴリズムなどにも興味がそそられます。連載では，これらのデータ構造やアルゴリズムをできるだけ詳細に解説していきたいと考えています。

連載は，まずF-BASICのメモリマップとワークエリアの紹介から始め，その詳しい解説・応用と進んでいきます。読者のみなさんの中には，F-BASICシステムの解析を志しながら，ディスアセンブルの際に，必要なデータ領域の分離に苦労している方も少なくないと思います。今号から3回に分けて紹介するメモリマップでは，データ領域とプログラム領域とを完全に分離していますので，ディスアセンブルを行う際，参考になると思います。

連載を始めるにあたって，読者のみなさんに1つだけお願いしておきたいと思います。

F-BASICシステムのROM内には，私たちが機械語でプログラムを作成する場合に有用なサブルーチンが数多く含まれています。今回の解析の目的の1つには，こうした有用なサブルーチンをできるだけ多く，正確に紹介することにあります。こうした有用なサブルーチンを個人的に利用する分には何も問題がありませんし，プログラミングの手間を省くためにも大いに利用していくべきだと考えます。しかし，ROM内のルーチンを活用したプログラムを市販する場合には，問題が起こりうるということが十分考えられます。例えば，バージョンアップなどによってROM内のサブルーチンのエントリアドレスが変更された場合には，それを使用したプログラムはたちまち暴走を始め，最悪の場合にはディスクのファイルを破壊しないとも限らないわけです(PCシリーズでは，この点が大問題となっています)。

富士通のFMシリーズでは，割り込み処理とRS-232Cを除くすべての入出力処理を，BIOSと呼ばれるI/Oドライバモジュールを経由して行うようにしていますが，その最大の理由の1つは，上記のようなことを避け，同時に機械語プログラムの機種間の互換性をできるだけ保つことにあるといえましょう。不特定多数のユーザの使用を目的にして作成するプログラムについては，以上の点からROM内のサブルーチンを使用することは，ぜったいに避けていただきたいと考えます。

最後に。今回の連載で分析の対象としたF-BASICシステムプログラムは，V3.0，82/09/20のものです。このデータは，$FBA5番地以降に，

```
30 33 30 30 38 32 30 39 32 30
 0  3  0  0  8  2  0  9  2  0
```

と書かれていますのでMONコマンドのDコマンドなどで確認してみてください。なお，BIOSについては，$F176番地以降に

```
56 01 4C 01 82 09 01
 V 01  L 01 82 09 01
```

とVersion Numberで書き込まれています。

メモリマップ （その１）

アドレス	内　　容
8000	拡張ステートメントポインタテーブルをセットする（ROMモード用）
8013～801C	拡張ステートメントポインタテーブル$01F9～$0202用データ（ROMモード用）
801D～8025	未定義中間コード用ステートメントデータ
8026～802F	拡張ステートメントポインタテーブル$0203～$020C用データ
8030～804D	中間コード$EE～$F3のステートメントテーブル(CHAIN, ERASE, LLIST, LPRINT, SOUND, PLAY)
804E～8059	上記ステートメントの処理ルーチンエントリポイントテーブル
805A	中間コード$EE～$F3のステートメントの処理ルーチンへ，分岐処理を行う。中間コードが$F3を超える場合には，拡張ステートメントポインタテーブル$020D～の処理ルーチンポインタ（相対値3～4番地）を参照し，ジャンプする（JMP〔$210〕）
8071～8074	中間コード$FFB4のステートメント(LPOS)データ
8075～8076	上記LPOSステートメントの処理ルーチンエントリポイントテーブル
8077	中間コード$FFB4のステートメント（LPOS）処理ルーチンへ分岐処理を行う。$FFB4を超える場合には，拡張ステートメントポインタテーブル$0212～の処理ルーチンポインタを参照しジャンプする（JMP〔$0215〕)。なお，この拡張ポインタテーブル$0212～は，関数の拡張用であり，使用にあたって注意すべき点がある。この点は，後に詳しく解説する
808E	CHAIN文メインルーチン。CHAINされるプログラムへ引き渡す変数の退避のほか，もし指定があればソースプログラムのDELETEを行ったのち，プログラムロードルーチンへジャンプする
80EB	CHAIN文の目的プログラムをロード後，呼び出される。ストリングスタックなど必要なワークエリアの初期化，退避させていた変数の目的プログラムへのリンクなどを行った後，目的プログラムを実行する
8131	CHAIN文のオペランド解析ルーチン
81A7	COMMON文をサーチし，見つかった場合，$81F1のサブルーチンをコールする
81F1	COMMON文で指定された変数について，変数エリアに格納されている変数名の頭文字のビット6をクリアする

アドレス	内　　容
8244	COMMON文で指定された変数が，変数エリアにあるかどうかチェックする
826F	FN関数をERASEする
827F	CHAIN文の目的プログラムに引き渡さない変数をERASEする
82FB	〔Y〕番地から〔X〕－1番地までの配列変数を削除
8309	〔$0061〕番地から〔$0049〕番地までのストリング配列変数の値をストリングストレージ内から削除
8313	変数名の次のアドレスを求める（入力：Xレジスタに変数の先頭番地。出力：Xレジスタに変数名の次の番地，Aレジスタに変数の型）
8321	文字変数の値がテキスト中にある場合に，それをストリングストレージにコピーする
835E	テキスト中の文字定数をストリングストレージにコピーし，正規化する（単純変数）
838E	同上（配列変数）
83A4	すべてのストリング変数を正規化する
83BF	CHAIN文実行中のエラー回復処理を行う。処理の内容は，バイナリファイルのMERGE，および退避中の変数の目的プログラムへのリンクの２つ
8448	ERASE文を実行する
848B	コールドスタート時のワークエリアなどのイニシャライズルーチン
8573	通信回線用ファイルコントロールバッファを設定する
8597	ブロック転送を行う（転送データ先頭番地…Xレジスタ，転送先先頭番地…Uレジスタ，転送データ数…Bレジスタ）
859F	AレジスタおよびXレジスタのデータを(U)番地から(B)組格納する
8656	コンソールを初期化する
8678～867A	キー先行入力禁止コマンドデータ
867B	BIOS初期化
8684	ホットスタート時の初期化ルーチン
8692	割り込みベクタをセットし，IRQをマスクする
86A6	タブを設定する（$02A2～$02AB番地までの10バイトに，タブ設定用の80ビットのビットイメージが書き込まれている）
86CC	PFキー文字列をセットする
8703～8758	ログインメッセージ用データ
8759～876A	COM0コントロールブロック用データ
876B～87B4	PF文字列データ

アドレス	内容
87B5～87C0	1文字読み出しプログラム。$00D2～$00DD番地へ転送される
87C1～87D2	システムインタラプトフックテーブル用データ($01D1～$01E2番地へ転送される)
87D3～87DE	$01E3～$01EE番地への転送用データ。文字色，背景色，UNLIST行番号などのデフォルト値
87DF～87E8	ステートメントテーブル用データ。$01EF～$01F8番地に転送される
87E9～87FD	LPT0コントロールブロック用データ。$0778～$078C番地に転送され，$06F2番地のポインタによって先頭が示される
87FE～8803	KYBD, SCRN, CAS0のファイルコントロールブロックポインタの値。$06EC～$06F1番地に転送される
8804～880F	割り込みベクタ用データ。$FFF2～$FFFD番地に転送される
8810～8815	DATE$のデフォルト値。YY/MM/DD=82/12/01。$02FA～$02FF番地に転送される
8816～886B	関数処理ルーチンポインタテーブル。$FF80～$FFAAの中間コードで表している33の関数用
886C～888F	演算子の演算優先順位と演算処理ルーチンポインタのテーブル。3バイトが1組になり+,－,＊,／,,^ AND, OR, XOR, EQV, IMP, MOD, ¥の順に，演算優先順位と処理ルーチンのエントリアドレスが並んでいる。ただし，"^"については，べき乗の処理ルーチンのエントリアドレスでなく，関係演算の処理ルーチンのエントリアドレスになっている
8890～8AD9	コマンド，ステートメントの予約語テーブル。中間コード$80～$E7までの$68 (104) 個のステートメント，コマンドおよび中間コード$FF80～$FFAAまでの$2B (43個) の予約語がコードの順に並べられ，それぞれの予約語は，最後の文字のビット7が立っていることにより区分されている。また，テーブルの最後は0によって知ることができる
8ADA～8B71	中間コード$80～$CBまでの76個のステートメント，コマンドの処理ルーチンエントリポイントテーブル
8B72～8D22	予約語のポインタと中間コードのテーブル。予約語のアルファベット順に並べられていて，頭文字の区切りは0によって知ることができる。中間コードは，ビット7が立っているものについてはそのままの値が，立っていないものについては，その値に$80が加えられたうえに，頭に$FFがつけられ2バイトの値として使用される
8D23～8D56	予約語ポインタ，中間コードテーブルのアルファベット順のインデックステーブル。予約語の頭文字順に予約語ポインタ，中間コードテーブルの先頭番地が示されている。なお，J, Q, Y, Zのようにこれらのアルファベットで始まる予約語がない場合には，値は$0000になっている。
8D57～8D60	"Ready" メッセージデータ
8D61～8D68	"Break" メッセージデータ
8D69	FOR～NEXT文, GOSUB～RETURN文のコントロールブロック先頭番地を求める
8D93	テキスト挿入用のメモリ領域を生み出す
8DAA	必要なメモリ空間が確保できるかチェックする。確保できない場合，Out Of Memoryエラーへジャンプ
8DBC	X←(B)+(X)
8DBD	X←(D)+(X)
8DBF	X←($0023・$0024)+(X)
8DC6	Out Of Memory エラー処理を行う
8DCF	Direct Statement In File エラー処理を行う
8DD1	エラー処理を行う(エラー番号をBレジスタに格納し，このルーチンに制御を移す)
8E63	Breakメッセージ，Abortメッセージを出力し，$8E72番地へ制御を移す
8E72	出力デバイスを"SCRN:"にセットした後，$8E82番地へ制御を移す
8E82	"Ready"メッセージを表示した後，$8E88番地へ制御を移す
8E88	コマンドまたはテキスト入力および編集メインルーチン
8F1C	テキストの検索を行う (検索する行の行番号を$004B・$004C番地に受け取り，レジスタおよび$0069番地にその行の先頭番地を戻す。指定した行番号がない場合にはキャリーフラグをセットして呼び出し元へ戻る
8F32	NEW コマンドエントリ (NEW1)
8F39	NEW2
8F4B	NEW3
8F51	NEW4
8F63	NEW5
8F70	NEW6
8F82	NEW7
8F91	NEW8
8FBC	RESTORE ステートメント実行ルーチン

アドレス	内容
8FC9	RESTORE 初期設定を行う
8FD0	END ステートメント実行ルーチン
8FD7	BREAK 処理を行う
8FD9	STOP 文の処理を行う
8F44	BASIC プログラム実行終了後処理を行う
9002	CONT コマンドの実行ルーチン
9012	RUN コマンドの解釈ルーチン
902D	GO エントリ
9039	GOSUB 文の処理を行う
9055	GOTO 文の処理を行う
9071	RETURN 文解釈実行ルーチン
90A0	DATA 文エントリポイント—テキストポインタをステートメントの終わりまで進める
90A3	ELSE 文，REM 文エントリポイント—テキストポインタを行の終わりまで進める
90A8	〝：″までテキストポインタを進める
90AB	行の最後までテキストポインタを進める
90AE	Bレジスタの値が見つかるまでテキストポインタを進める
90F1	IF 文解釈実行ルーチン
9134	ON（式）GOTO, ON（式）GOSUB, ON ERROR GOSUB エントリ
913A	ON(式)GOTO または ON(式)GOSUB の処理を行う
9162	ASCII 10 進数をバイナリに変換する（符号なし 2 バイト整数）
9195	代入文（LET 文）メインルーチン
91B3	文字列の代入を行う
91E8	Missing Operand エラーエントリ
91ED	単項式の評価を行う
9288	（式）の評価を行う
928C	右カッコ〝)″チェックを行う
928F	左カッコ〝(″チェックを行う
9292	コンマ〝,″チェックを行う
9294	Bレジスタの値のコードのチェックを行う
92A0	Syntax Error エントリ
92A5	〝−″の単項演算を行う
92BE	式の評価を行い，文字式の場合には Type Mismatch エラー処理を行う
92C3	式の評価を行う
9330	2 項演算を行う
9364	FAC1 の値と FAC2 の値とを交換する（倍精度）

アドレス	内容
9374	同上（単精度）
9384	同上（整数）
939E	指数部を FAC1 に，被演算数を FAC2 に格納して，べき乗演算ルーチンへジャンプする
93B0	論理演算エントリ
93BB	¥，MOD 演算エントリ
93CD	除算エントリ
93E8	FAC1 の値をスタックに積む
93F4	同上（FN 関数処理ルーチンが呼び出す）
9411	スタックの値を FAC2 へ戻す
9433	関係式演算メインルーチン
946B	文字型の値の関係式演算を行う
94A8	AND 演算ルーチン
94AD	OR 演算ルーチン
94B2	XOR 演算ルーチン
94B7	EQV 演算ルーチン
94BC	IMP 演算ルーチン
94C5	DIM エントリ
94CE	変数名を$0568～$0577 番地のストレージに，変数名の長さを$0057 番地に格納する。変数名は最大 16 文字までに制限される
94FD	レジスタAの文字が英大文字かどうかチェックする。英大文字の場合，キャリーは 0，ほかは 1
9506	レジスタAの文字が数字かどうかチェックする。数字の場合，キャリーは 0，ほかは 1
950F	変数が定義済みかどうかのチェック，変数の定義，変数の値の格納番地を求めるなどの処理を行う（詳細は連載中に解説する）
95AD～95AE	ジョブ選択用データ。上記ルーチンを呼び出しているプログラムを特定するためのもの
95AF	変数名の照合を行う
95C7	変数パラメータを生成する
95D9	テキストポインタを 1 バイト進めた後，式の評価を行い 1 バイトの正の整数を得る
95DB	テキストポインタの更新を行わないほかは上に同じ
95DE	フローティングアキュムレータ 1（FAC1）の値を 1 バイトの正の整数に変換する
95E6	配列変数の検索を行う
9660	Subscript Out Of Range エラーエントリ
9663	Illegal Function Call エラーエントリ
966E	配列変数を登録する

アドレス	内　　容
96BC	配列変数名で指定されるデータが格納されている先頭番地を求める
96F3	Dレジスタの値と$008D：$008E番地の値の積を求め，結果をDレジスタに持ち帰る（ただし，結果が2バイトを超える場合にはエラー）
972C	FRE関数の値を求める
974C	FAC1に正の整数（0～255）を格納する
974D	FAC1に符号付き整数を格納する
9752	STR$関数の処理を行う
9760	ストリング列の長さを求める 入力：ストリング先頭番地…Xレジスタ 　　　ストッピングデリミタ…Dレジスタ 出力：ストリングの長さ…$007E番地 　　　ストリングの先頭番地…$007F番地
9790	ストリングストレージ領域内にストリング変数をコピーする領域を確保した後，ストリングの長さを$007E番地に，先頭番地を$007F番地に格納して戻る
9799	ストリング定数の評価を行う
979B	同上
979D	同上
979F	同上
97AD	ストリングスタックにストリング定数データを格納する
97D7	ストリングストレージ内に，文字列をコピーする領域を確保する
980D	ストリングストレージ内のガベージコレクションを行う
9850	ストリングストレージ内の文字列の格納領域をシステムに返す
9865	ストリングストレージ内の文字列を正規化する
987C	ストリング配列変数を正規化する
98AE	ストリング変数の和を求める
98E4	文字列をストリングストレージにコピーする
98E8	同上
98F1	文字列の先頭番地と長さを求める
991B	ストリングスタックを1個解放する
992A	LEN関数の処理を行う
9933	CHR$関数の処理を行う
9947	ASC関数の処理を行う
9952	LEFT$関数の処理を行う
996F	RIGHT$関数の処理を行う
9979	MID$関数の処理を行う

アドレス	内　　容
999C	LEFT$関数，RIGHT$関数，MID$関数の第1パラメータ，第2パラメータの値をスタックから取り出す
99BA	TEXTポインタを次の文字の位置に進めた後，式の評価を行い，FAC1に正の整数（0～255）を得る
99BC	TEXTポインタの更新を行わないほかは上に同じ
99BF	FAC1の値を正の整数に変換する（ただし，値が256以上の場合には，Illegal Function Callエラー）
99CA	VALエントリ
99F9	〝,〟で区切られた第1パラメータ（2バイト符号なし整数），第2パラメータ（1バイト符号なし整数）を評価し，結果を$004B：$004C番地およびBレジスタに格納して戻る
99FD	〝,〟チェックの後，式を評価し，Bレジスタに1バイトの符号なし整数を得る
9A02	式の値を符号なし整数として評価し，結果をXレジスタに格納して戻る
9A05	FAC1の値を符号なし整数に変換し，その値をXレジスタに格納して戻る
9A22～9A25	65536_{10}の単精度実数値のデータ
9A26	PEEK関数の処理を行う
9A30	POKEエントリ
9A3A	行番号定数を評価し$004B番地に格納する
9A50	TAB(　)エントリ
9A7F	LPRINTエントリ
9AB1～9AB2	BIOS LPCHKコマンド用データ。$1700
9AB3	PRINT文エントリ
9AC0	PRINT文，LPRINT文実行ルーチン
9B47	各出力デバイスに対してニューライン出力処理を行う
9B50	ニューライン出力ルーチン
9B68	PRINT文，LPRINT文中の〝,〟処理を行う
9B7E	PRINT文，LPRINT文中の〝;〟処理を行う
9B9F	出力デバイスがプリンタかどうかをチェックする
9BB5	SPC(　)関数の処理を行う
9BD4	Bレジスタの値だけスペースを出力する
9BDB	1行出力ルーチン（[X]+1番地から$0で終わるデータ）
9C02	SCRNへ1行出力する（出力データ先頭番地Xレジスタ，出力データ数Bレジスタ）
9C20	〝?␣〟出力
9C22	〝␣〟出力
9C25	〝?〟出力

アドレス	内容
9C2B	PRINT@文の処理を行う
9CD4	INSTR 関数の処理を行う
9D59	STRING$関数の処理を行う
9D7B	SPACE$関数の処理を行う
9D91	ストリング文字列がテキスト中にある場合，それをストリングストレージにコピーする
9DC1	代入文の左辺に書かれたMID$文の処理を行う
9E2C	HEX$関数の処理を行う
9E2D	OCT$関数の処理を行う
9E7E	＆定数の処理を行う
9E94	&O 定数の値を求める
9EA5	&H 定数の値を求める
9ECC	$0048，$0049 番地の2バイトの値を，10進ASCII文字列に変換する
9ED1	BASICプログラムをASCII形式で出力デバイスに出力する
9F0B	LIST コマンド，DELETE コマンドなどのオペランドの行番号を絶対番地に変換する
9F62	行番号の値を$004B番地のTemp Storageに格納する
9F73	DEFSTR エントリ
9F76	DEFINT エントリ
9F79	DEFSNG エントリ
9F7C	DEFDBL エントリ
9FBC	DELETE エントリ
9FD3	テキストの〔$00A2〕番地から〔$00A4〕－1番地までを削除する
9FE7	AUTO 文の処理を行う。先頭行番号を$009E 番地に，増分値を$00A0 番地に格納し，エディタルーチンヘジャンプする
A01A	TRONコマンドの処理。$00A9番地に$4Fを格納する
A01B	TROFF コマンドの処理。$00A9 番地をクリアする
A01F	PRINT USING〝！〃文の処理を行う
A023	PRINT USING〝@〃の処理を行う
A058	PRINT USING〝&　　〃の処理を行う
A07F	PRINT USING 文エントリ
A203	FOR 文エントリ。制御変数の格納番地，増分値，最終値などのパラメータをスタックに格納する
A2C7	NEXT 文エントリ
A36D	WHILE～WEND ループのネストのチェックを行う
A3EF	文字定数と数値定数をスキップし，次のコードを得る
A410	REM 文，'文，DATA 文のとき，Zフラグをたてると

アドレス	内容
	ともにテキストポインタを行の終わり（REM 文および'文）またはセパレータ（DATA 文）まで進める。数値定数の場合には，Zフラグはクリアされるが，テキストポインタは次のコードまで進められる。他のトークンに関してはZフラグがクリアされるほか，テキストポインタの変更はされない
A43F	現在のテキストポインタの1つ手前のコードを得る（PRINT USING 文処理ルーチンの一部）
A446～A70E	エラーメッセージデータ
A70F	ON ERROR GOTO 文の処理を行う
A739	ERROR 文エントリ
A744	RESUME 文エントリ
A784	SWAP 文エントリ
A7C4	ストリングストレージ内の文字列のリンクポインタを付け替える
A7D9	DEF 文エントリ
A7DF	DEF FN 文の処理を行う
A80E	FN 関数の検索を行う
A81F	RUN モードのチェックを行う
A825	Illegal Direct エラーエントリ
A82A	FN 関数の評価を行う
A892	（$00AA：$00AB）↔（$00D9：$00DA）
A925	文字型引数を評価する
A952	VARPTR 関数の処理を行う
A969	DEFUSR 文の処理を行う
A980	USR 関数のポインタアドレスを求める
A9A0	USR 関数の処理を行う
A9CA	RENUM コマンドエントリ
AA2E	変更後の行番号が制限内（63999）に納まるかどうかをチェックする（RENUM PASS1）
AA2F	行番号を付け替える（RENUM PASS3）
AA4E	行番号テーブルに登録済みかどうかをチェックする
AA6C	行番号テーブルへの登録を行う
AA86	行番号テーブルの拡張を行う
AAA8	行番号テーブルを作成する（RENUM PASS2）
AB73	行番号の付け替えを行う
AB9A	未定義行番号があるかどうかチェックする（RENUM PASS4）
ABF4	MON コマンドメインルーチン
AC2F	Gコマンド処理ルーチン
AC37	Dレジスタの値を16進ASCIIコードに変換し出力デ

アドレス	内容
	バイスに出力する
AC3D	Aレジスタの値を16進ASCIIコードに変換し，出力デバイスに出力する
AC54	ASCII 16進コード1バイト入力。入力した値はバイナリコードに変換されBレジスタに保持される
AC5F	キーボード1行入力ルーチン
AC6C	ASCII 16進コード2バイト入力ルーチン。入力した値はバイナリコードに変換されDレジスタに保持される
AC95	Mコマンド処理ルーチン
ACC4	Dコマンド処理ルーチン
ACFB	Rコマンド処理ルーチン
AD36～AD46	レジスタ名データ
AD47	レジスタ名出力ルーチン
AD54	WHILE文エントリ。WHILE～WEND文の処理に必要なポインタなどをスタックに格納する
AD81	WEND文エントリ
ADDC	HARDC文エントリ
ADF0	HARDC0文の処理を行う
AE69	HARDC1文の処理を行う
AE8C	HARDC2文の処理を行う
AEBD～AED1	印字カラー指定バイト用データ
AED2	ANPORT文エントリ
AF05	実数の加減算を行う
AF0C	FAC1←FAC1＋0.5
AF11	FAC1←F(X)－FAC1
AF14	FAC1←FAC2－FAC1
AF1A	FAC1←F(X)＋FAC1
AF1D	FAC1←FAC2＋FAC1
AF93	FAC1の値の正規化1(もしキャリー＝0の場合,FAC1の値の2の補数をとった後に正規化を行う)
AF97	FAC1の値の正規化を行う
B011～B043	FAC1←－(FAC1)
B044	Overflowエラーエントリ
B049	フローティングアキュムレータの仮数部の右シフトを行う
B089～B08C	単精度実数1.0のデータ
B08D～B0A9	LOG演算用定数などのデータ
B0AA	LOG関数エントリ
B0EB	FAC1←0.693147×FAC1
B0EE	FAC1←FAC1×F(X)
B0F6	FAC1←FAC1×FAC2

アドレス	内容
B1BA	〔X〕番地の実数をFAC2に格納する
B1DF	FAC1とFAC2の指数部の加算を行い，結果をFAC1の指数部に格納する
B206	FAC1←FAC1＊10.0
B219～B220	10.0の浮動小数値
B221	Division By Zeroエラーエントリ
B226	FAC1←FAC1/10.0
B234	FAC1←F(X)/FAC1
B236	FAC1←FAC2/FAC1
B2EC	FAC3の仮数部をFAC1の仮数部に格納する
B301	X番地に格納されている数値定数をFAC1に格納する
B32B	FAC1の値を$0063番地から始まるワークエリアに退避する
B330	FAC1の値を$005F番地から始まるワークエリアに退避する
B335	FAC1の値を$005A番地のポインタが指している番地に退避する
B337	FAC1の値をXレジスタの値の番地に退避する
B35F	FAC1の値をFAC2へ代入する
B380	FAC2の値をFAC1へ代入する
B3A0	FAC1の値をチェックし，0の場合に0を，負の場合に－1を，正の場合に1をBレジスタに格納して戻る(実数)
B3A4	FAC1の符号をチェックし,正の場合に1を,負の場合に－1をBレジスタに格納して戻る
B3A6	Bレジスタの値を符号付き2進数とみて，正の場合に1を,負の場合に－1($FF)をBレジスタに格納して戻る
B3BE	SNG関数の値を求める
B3C0	Bレジスタの値を整数化し，FAC1に格納する
B3C8	FAC1の値の符号をチェックし，0の場合に0,負の場合に－1を,正の場合に1をBレジスタに格納して戻る
B3D5	ABS関数の値を求める
B3E1	FAC1とFAC2の値の比較を行う
B3EC	FAC1とF(X)の値の比較を行う ①
B3F6	FAC1とF(X)の値の比較を行う ②
B435	FAC1の少数以下の値を切り捨てる
B461	FIX関数の値を求める
B472	INT関数の値を求める
B506	ASCII文字列の数値を倍精度実数に変換し，FAC1に格納する
B50B	ASCII文字列の数値を2進数に変換し，FAC1に格納

アドレス	内容
	する
B609～B60D	"␣IN␣" データ
B60E	"␣IN␣行番号" を表示する
B615	Dレジスタの値を符号なし整数として10進ASCIIキャラクタに変換し，出力
B620	FAC1の値をASCII文字列に変換し，出力バッファに格納する（フォーマット指定なし）
B622	FAC1の値を，ASCII文字列に変換し，出力バッファに格納する。その際フォーマット書式制御の指定があれば，その指定に従う
BA62～BAED	数値ASCII文字列変換用データ
BAEE	SQR関数の値を求める
BAF7	べき乗の計算を行う
BB36～BB5A	EXP関数演算用データ
BB5B	EXP関数の値を求める
BBE5	RANDOMIZEエントリ
BBFE	RND関数の値を求める
BC4C～BC4E	RND関数の種のデータ
BC4D～BC73	RANDOMIZEキー入力要求プロンプトメッセージ用データ
BC74	CINT関数の値を求める
BCA5	文字変数チェック
BCAE	数値変数チェック
BCB3	変数の型を調べる
BCC1	変数の型をAレジスタの値が示す型に変換する
BCCD	CDBL関数の値を求める
BCD5	単精度実数を倍精度に変換する
BCE0	CSNG関数の値を求める
BCEE	倍精度実数を単精度に変換する
BCFF	FAC1の符号付き整数を単精度実数に変換する
BD01	Xレジスタの値を符号付き整数とみて単精度実数に変換する
BD19	FAC1の整数を符号なしとみなして単精度実数に変換する
BD29	加減乗算および関係演算ルーチンで呼び出され，演算数が整数か実数かをチェックした後，実数の場合には，実数演算ルーチンへジャンプする
BD3F	FAC1←(FAC1)−(FAC2)
BD44	FAC1←(FAC1)(整数)−(FAC2)(整数)。ただし，演算の結果が，整数の範囲を超えた場合には，FAC1，FAC2の値を実数に変換した後，減算を実行する

アドレス	内容
BD4D	FAC1←(FAC1)+(FAC2)
BD52	FAC1←(FAC1)(整数)+(FAC2)(整数)。ただし，演算の結果が実数の範囲を超えた場合には，FAC1，FAC2の値を実数に変換し乗算を実行する
BD5C	FAC1，FAC2の値を実数に変換する
BD73	FAC1←(FAC1)*(FAC2)
BD78	FAC1←(FAC1)(整数)*FAC2(整数)。ただし，演算の結果が整数の範囲を超えた場合には，FAC1，FAC2の値を実数に変換し乗算を実行する
BDC9	FAC1←−(FAC1)
BDCE	FAC1←−(FAC1)(整数)，ただし，結果が32768の場合には実数に変換する
BDD9	FAC1，FAC2の符号付き整数の符号の乗算と値の絶対値への変換を行う
BE0C	FAC1←(FAC1)¥(FAC2)
BE33	FAC1←(FAC1)MOD(FAC2)
BE44	FAC1の値とFAC2の値を比較し，(FAC1)=(FAC2)のとき0を，(FAC1)<(FAC2)のとき$FFを，(FAC1)>(FAC2)のとき$1をBレジスタに持ち帰る
BE60	COS関数の値を求める
BE66	SIN関数の値を求める
BEB1	TAN関数の値を求める
BED4～BEF4	COS, SIN, TAN関数演算用データ
BEF5	ATN関数の値を求める
BF23～BF47	ATN関数演算用データ
BF48	LINE INPUTエントリ
BF71	INPUT文のオペランド処理
BF72	LINE INPUT文のオペランド処理
BFB0	INPUT, LINE INPUT 1行入力ルーチン
BFC0	INPUT入力エラー処理ルーチン
BFDD	INPUT文の処理を行う
BFF3	READ文の処理を行う
C156～C168	"?Redo From Start" プロンプトメッセージ
C169	BASICプログラム1行変換ルーチン（バイナリコード→ASCIIコード）
C288	BASICプログラム1行変換ルーチン（ASCIIコード→バイナリコード）
C590	英小文字→英大文字変換および英字チェックルーチン
C595	英小文字→英大文字変換ルーチン
C5E2	関数（中間コード＄FF80～＄FFAA）のパラメータ処理および実行ルーチンへの分岐処理を行う

プレゼント情報

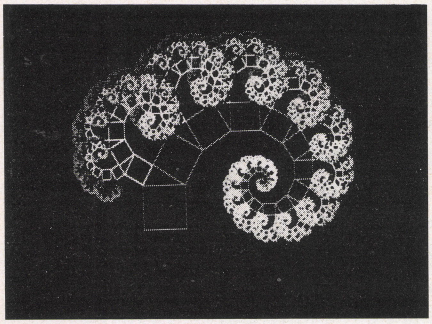
今月号のプレゼントを紹介します。ご希望の方はとじこみのアンケートはがきに必要事項記入のうえ，お送りください。しめ切りは4月15日(必着)。当選者発表は6月号（5月18日発売）で行います。

❶ FM-Logo 富士通提供

FM-7用テープ版 5名

FM-7用ディスク版 3名

希望媒体(TまたはD)を明記のこと。



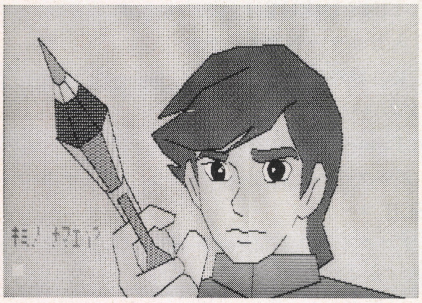
❷ F-BASIC86（2D） 3名

富士通提供

❸ 四次元少女リディア（FM-7用） 5名

チャンピオンソフト提供

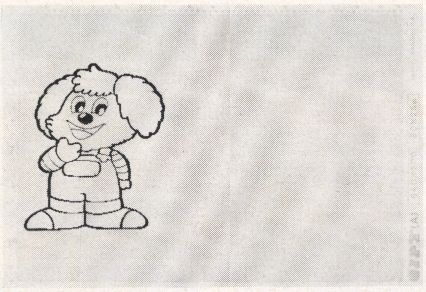
6号SOFT REVIEWで紹介済みだが，自分自身が主人公となる新しいタイプのSFアドベンチャゲーム。プレイヤはステレオグラスをかけて，飛び出す立体映像を楽しめる。定価は4,800円。



❹ GIPY 25名

サンシュープログラムアーツ提供

640×200ドット用のデザインシート。写真やイラストを1ドット単位でレイアウトできるすぐれたもの（本誌5号の付録と同じだけど，トレーシングペーパーに印刷されている)。定価は30枚入1000円。



❺ 今月号掲載のマシン語モニタ

テープ版 30名

ディスク版 10名

73ページに掲載のマシン語モニタです。

❻ Oh！FMトレーナ L，M各10名

サイズ明記のこと。クリームイエローの地に茶色で「Oh！FM」と入ってます。

❼ Oh！FM 4月号ノート

本誌と同じ体裁・表紙で，中はなにも印刷されていないメモ帳です。

7号のプレゼント当選者

（敬称略）

①MB27607（1名）
神徳宏（福岡市）

②PIPE MAN（5名）
服部和夫（大阪市）・渋谷賢司（益田市）
河本隆文（足立区）・稲田健（世田谷区）
寺上やよい（筑紫野市）

③サイン色紙（6名）
金敦也（岩見沢市）・山崎実（埴科郡）
ほか4名

④Oh! FM Vol.7ノート（10名）
石田勝嗣（新潟市）・百瀬博文（藤沢市）
川村俊哉（千歳市）・秋元克紀（江東区）
池田一仁（大田区）・鈴木沅弘（清瀬市）
高岩聖悟（大阪市）・宮下恵子（東国東郡）・裏山隆一（東大阪市）・宮嵜重利（上水内郡）

⑤Oh！FMシャープペンシル（50名）
太田信也（多摩市）・岩辺卓也（清水市）
橋川健竜（横浜市）・加藤光一（藤沢市）
横山哲也（京都市）・荻井利宏（中野区）
露木幹高（横浜市）・内山次郎（千葉市）
田中洋司（昭島市）・山本和幸（札幌市）
ほか40名

オリジナルプログラム募集

1オリジナルプログラム

2楽符プログラム

歌謡曲，クラッシックなど音楽のジャンルは問いません。FM-7用，FM-X＋FM-7用など使用機種を明記してください。

両方とも①プログラムの内容，特徴，アルゴリズムなどの説明，②フローチャート，変数表，③遊び方や操作方法，注意事項，④プログラムの入ったテープないしはディスケット（氏名明記のこと)，⑤機械語の場合はソースプログラム添付，以上を住所・氏名・年齢・職業・電話番号などを明記のうえ，郵送するか直接お持ちください。

あて先は
〒102 東京都千代田区四番町2-1
㈱日本ソフトバンクFM投稿係

- プログラムは未発表のものに限ります。
- 他誌との二重投稿，盗作などモラルに反する応募は固くお断りします。
- 応募原稿は原則としてお返し致しません。
- 掲載にあたって多少修正させていただくことがあります。
- 応募者全員にOh！FMシャープペンシルを進呈。
- 掲載の場合は本誌規定の原稿料をお支払いいたします。

読者の皆さまのご意見をお寄せください。誌面づくりに反映させるための貴重な資料となりますので、ご協力ください。

料金受取人払

郵便はがき

102

麹町局承認
161

差出有効期間
昭和60年3月
14日まで

(受取人)

東京都千代田区
四番町2-1

㈱日本ソフトバンク

Oh!FM 編集室行

プレゼントNo.		整理番号	*

氏名	年齢	性別 男　女	マイコン歴 年　か月
住所 〒　都道府県 ☎			
職業・勤務先 学校・学部・学年			
本誌の購読は	はじめて, 第　号から ときどき, 毎号	本誌以外によく読む雑誌	

*印の欄には記入しないでください。

愛読者アンケート

○特集について
非常によかった　よかった　普通　よくなかった
(理由　　　　　　　　　　　　　　　　　　　　　　　　　　　)

○本誌をご覧になった感想

○いちばん良かった記事

○興味のなかった記事

○載せてほしい記事・内容を具体的に

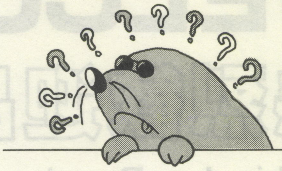
○どんなパソコンをお持ちですか　(個人用　会社等　ない)
FM-7　-8　-II(EX, AD, ST, BS, AD2)　-X　他機種
CRT<グリーン・カラー>(メーカー名　　　　　　　　)／家庭用
Dレコーダ(メーカー名　　　　　　　　)／ラジカセ
ディスク<S・D>(型番 メーカー名　　　　　　　　)
プリンタ(型番 メーカー名　　　　　　　　)
漢字ROM　Z-80カード　RS-232C　PSG　バブル
その他(　　　　　　　　　　　　　　　　)

○今，一番気にいっている市販ソフト(発売元も)

○あなたがソフトを買う時どのような基準で買いますか，またその情報はどこから知りますか。

読者の皆さまのご意見をお寄せください。誌面づくりに反映させるための貴重な資料となりますので、ご協力ください。

Question & Answer



Q Oh!FM第5号のショートプログラム"スカッシュ"の行番号1010を，RANDOMIZE TIME ＊.75-32768としたところ，このあとに！がついてきました。これはなぜついてくるのですか。また，！がついていて正しいのですか。(松戸市・中村卓)

A この！という記号は，実は定数や変数の型を表しているもので，気に入らないからと言って[DEL]キーなどでエディットして消しても，リストを再度とってみると，しっかりとまた付いてしまいます。F-BASICで扱う数値定数には，整数，単精度実数，倍精度実数，8進定数，16進定数のように型式があり，それぞれ%，!，#，&O，&Hという記号をつけることで区別しています。BASICのテキスト上で書き込まれた数値になんの記号も付いていなければ，F-BASICはそれを整数として処理しようとします。けれども，整数として処理できないものには，適切な型式指定の記号を付けて表示してくれるようになっているのです(逆に，整数の後にいくら%を付けてもリストをとると消えてしまいます)。F-BASIC上での整数というのは−32768から32767までの範囲しかありません。中村さんの使った32768という数は，惜しくも整数の範囲からはみ出してしまったため，単精度実数と判断されてしまい，！マークが付いたのです。"こんな面倒なことをしないで全部倍精度実数で処理すればいいじゃないか"と思う人もいるかもしれませんが，整数はメモリに格納するとき2バイトで済みますが，単精度実数で4バイト，倍精度実数では8バイトのメモリを必要とします。これはメモリの無駄使いであるばかりでなく，実行速度を下げる原因にもなってしまうのです。

Q マニュアルのシステム仕様や機械語関係の本に出てくる「リード」「ライト」が，どういうことかわかりません。例えば「$FD05のビット7をREADする」とは，具体的にどうすればいいのでしょうか。いろいろな本にあたってみましたが理解できず……Oh!FMだけが頼りです。(文京区・小泉要)

A 英語辞典を引いてみてください。「READ」と「WRITE」の意味は「読む」「書く」です。とは言っても，文字を読み書きするわけではありません。「メモリ上の値を読む」「メモリ上に値を書く」のです。BASICならPEEK，POKEなど，機械語ならLDA，LDXなどのロード命令が「READ」，そしてSTA，STDなどのストア命令が「WRITE」に相当します。「$FD05のビット7をREADする」というのは，

BASICなら　A＝PEEK（&HFD05）
機械語なら　LDA　$FD05

というようになります。なお，6809には1ビット単位で読み書きする命令はありません。

Q コンピュータグラフィックなどをVTRに録画したいと思い，グリーンCRTケーブルのコンポジット出力をVTRの映像端子に入れてみたところ，同期はするのですが白黒になりました。カラーで録画するにはどうすればよいでしょうか。CRTインタフェース部の回路を少し変更すればよさそうに思いますが。(奈良市・佐藤宗雄)

A コンポジット出力端子から出ているのはまぎれもない白黒の信号なのです。マニュアルをよく見てください。なお，CRTインタフェースを改造するとしたら，少しの変更どころではすみません。"RGB→コンポジットコンバータ"というものが数社より市販されておりますので，カラーでVTR録画をしたい方はそちらをご利用ください。

Q 先日FM-7用のMB27607薄型ミニフロッピを購入しましたが，使用機種がFM-8のため$6FB0〜$7000番地までをフロッピのDOS(V.2)に使用され，従来のプログラムが作動しません。$7000〜$8000番地の間が未使用なのでDOSのプログラムをそこに入れたいのですが，その移動方法を教えてください(ユーザエリアを$1000〜$7000にしたいという意味です)。(仙台市・蛯名康徳)

A DOS(V.2)とはなんのDOSでしょう？　はがきを見た限りでは理解できませんでした。おそらくF-BASIC V2.0のDOSだと思いますが，それでしたらちょっと勘違いをしておられるようです。F-BASIC V2.0では$6F64〜$71D5番地までの間にシステムのイニシャライズルーチン(実際に$6800番地からLOADされる)がありますが，このサブルーチンは一度使ってしまうと後は不要となりますので，フリーエリアにされてしまいます。すなわち，F-BASIC V2.0ではドライブ数2，ファイル数1(リターンキーのみ押した場合と同じ)を指定した場合でも$0CF3~$71D4番地までの間がユーザエリアとなるのです。なお，$71D6〜$8000番地は，しっかり使用されています。また，BASIC内のサブルーチンを使うようなマシン語ルーチンを持たないものならば，V1.0用のプログラムはすべて動くはずです。

＊　＊　＊

FMに関する質問をお寄せください。質問内容は文書で，できるだけ具体的にお書きください。ソフト・ハードなど幅広い質問をお待ちしています。ただし，マニュアルを読めばすぐわかるような質問はご遠慮ください。住所・氏名・電話番号などを明記のうえ，Oh!FM編集部Q&Aあてお送りください。

なお，質問に対する回答は誌上のみです。往復はがきなどによる質問はご遠慮ください。

WELCOME TO LOGO
熱烈歓迎呂昊 ようこそFMロゴへ

Thabio La Baugh

世界にはいろいろな人がいて，いろいろな言葉をしゃべっているように，コンピュータの世界にもいろいろな言葉が存在する。そして，言葉が違えば考えも違うように，コンピュータ言語にも文化圏のようなものがあるんじゃないかと思う。もし君がアフリカの文化を吸収したいと思ったら，まずどうする？ 古代ギリシャの文化に接してみたいと思ったらどうする？ そりゃパスポートを取っちゃうだろう。予防接種をしたりもするかもしれない。

でもまずしなくちゃいけないことは，その国の言葉を覚えることだよね。同じ言葉をしゃべっている人は，1つの文化をつくりあげる。ある言葉を使うってことは，ものの考え方を規定しているんじゃないかと思うほどだ。そしたらやっぱり，いろいろな考え方を知るためにいろいろな言葉を覚えたいと思うよネ。コンピュータ言語にしたってそうだ。

BASICを覚えたら次はFORTRANを覚えるのもいいけど，それじゃあ1つの文化圏を出てはいない。やっぱりLogoとかLISPとかprologなんかがかたちづくっている新しい文化圏が，今一番おもしろい。その中でもLogoは，構文面から，BASICなどの文化圏との橋渡し的存在なので，ごく自然に入っていくことができるというメリットがある。もしあなたが，BASICも何も知らない，コンピュータは全くの初めて，素人さんっというんだったら，それはとても幸運。

一度Logoを知っちゃうと，どうしてもBASICを使わなくちゃいけないときに，その不合理さに，腹がたってくるほどだから。Logoから始めるってことは，実用という面からは遠いけれど，頭脳のためには一番良い方法だと思う。

で，そろそろ実際に，この講座をどのように進めていくかを説明しよう。はじめはLogo入門ってことで引き受けたんだけど，なんといっても天下の富士通だからLogo 1つ売るにも，プログラミング入門とリファレンスマニュアル，合わせて350ページがついてくる。さっすが～!! リファレンスマニュアルに至っては，もう並のリファレンスマニュアルじゃあない。プログラミングのヒントがそこいら中にちりばめられてあるっていうすぐれたもの。全部読み切れば君はもう，Logoのプロフェッショナルだ！これだけ書かれていると，もう書くことがない。でも書かなきゃおまんまは食い上げだ，女は逃げるわ，病気にはなるわ，放火はされるわ，地震は起こるわ，もうパニックのタワーリングインフェルノ，そこで，〝三歩下がって 富士通の影を踏まず〟Logo入門はやめて，プログラミング入門とリファレンスマニュアルをある程度読んだと想定して，ちょっとわかりにくいんじゃないかという点を説明したあとLogo的なプログラミングをするのに必要な概念を扱ってみようと思う。

というわけで，今後の予定は，

- データの型，ワードとリスト
- 再帰的っていうこと
- EDITOR
- 自由変数と束縛変数
- LISP Logo prolog
- Logo システムの全容

といった具合に進めていきたいのだが，うまくいけばLogoコンパイラってやつを作ってもみたいとも思っている。

ワードとリスト

LogoがBASICやFORTRANと決定的に違うのは，扱っているデータが2進木リストという形をしていることによるものだ。世の中にはたくさんの言語があるけれど，言語の設計で一番苦労するのは，どのような形式でデータを扱うか，ということなんだ。とくにCOBOLなんて，ファイリングするためのデータと直結した言語だし，FORTRANは科学技術用の，精度の高い数値データを扱う。BASICは対話性を高めるため，強力な文字データの形式を持っている。でも知能の研究を扱うためには，数値や文字のデータが扱える，というだけじゃ不完全だよね。そこで2進木なんだけれど，これは，データとデータの間の関係づけを表すデータ形式，メタデータとでも言えばいいかな？ そんなものなんだ。じゃあその2進木ってやつを少し説明してみよう。

まず，2進木を含む木構造ってものから説明しなくちゃなんない。例として三目並べなんてのはどうだろう。

図1では，一番上の何も打っていない盤面を根として，ちょうど，木をひっくり返したような形をしている。実際に何か物事を考えるとき，本当に必要なのはデータそのものじゃなく，データとデータの関連性だろう。それを表すためにはどうしてもこのようなデータに対してメタの記述をする必要がある。

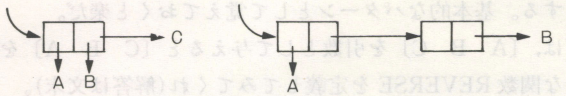
ところで，このような構造をもつ情報は非常に多いのだが，このままでコンピュータに処理をさせるには，ちょいと無理がある。そこで2進木で表すという方法が考えられた。データ形式を一意に決めておいて，処理すべき情報をその形に翻訳しようってわけだ。さて2進木だけれど，図2のような形をしている。

図中の□□はノードといい，→はポインタという。

2進木は次のような特徴をもっている。

- ノードからは2本のポインタが出ている（1本でも，3本以上でもいけない）。

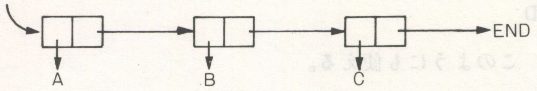
したがって次のようなものは許されない。



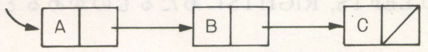
実際のLogoのデータは，どのような2進木になっているかをまずみてみよう。

PRINT　〔A　B　C〕

としたとき，〔A　B　C〕というリストはコンピュータ内部で2進木で表されている。



Logoでは，データの1つ1つはワードという呼び名で，その関係の表現をリストと呼ぶ。各々のA, B, Cはワードである。ENDというのはワードではなく，2進木の終わりを示すマークのようなものだと思ってほしい。上記のような2進木をこれから，



と簡略化して書くことにする。

〔〔A　B〕〔C　D〕〕

それでは，上のリストはどのように2進木として表されるのだろうか。答えは文末。

ワードっていうのはLogoの持つ大きな特徴で，たぶんBASICの文字列操作にヒントを得たものだと思うんだけど，これのおかげでLogoはとても使いやすくなっている。リストと同じ感覚で文字列を扱えるから構文も同じに書ける。

BUTFIRST BUTLAST FIRST LAST ITEM ENPTYPは，構文的にリストと文字列を同様に扱える強い味方だ。具体例は，次の「再帰的っていうこと」に載せるから，そちらを参照してね。

＝再帰的っていうこと＝

再帰的っていうことは，慣れれば簡単なんだけど慣れるまでがたいへん。なんでこんなまわりくどい言い方するの！　なんて怒る人もでてくる。

でも再帰的にしか書けないこともあるし，再帰的に書けば短くわかりやすいってこともよくある。とにかくLogoの本質を知るには，再帰的ってことなしには考えられないんだ。

NのM乗を計算するプロシジャを作ってみてくれ。いろいろな方法が考えられるけど，こんなふうに作った人が多いと思う。

図1

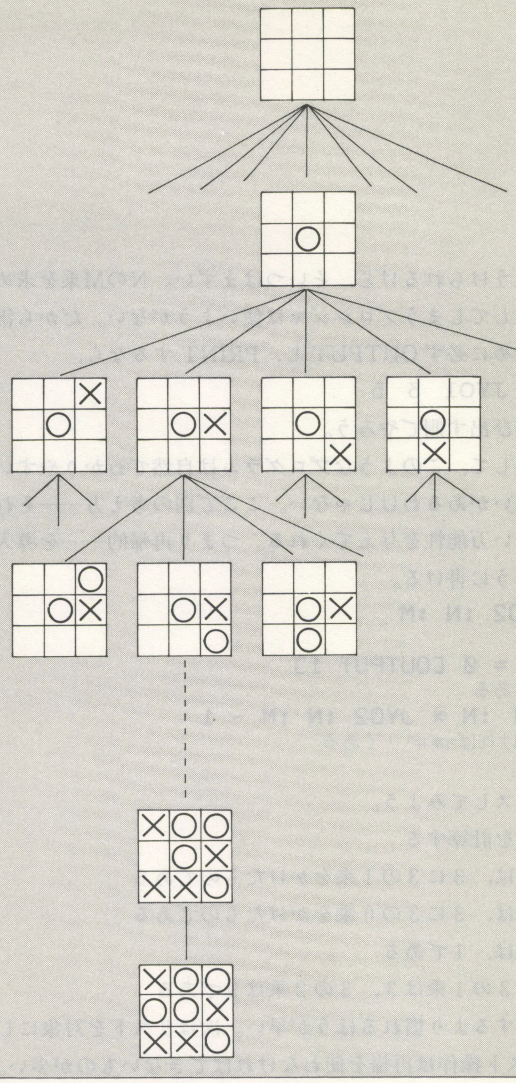
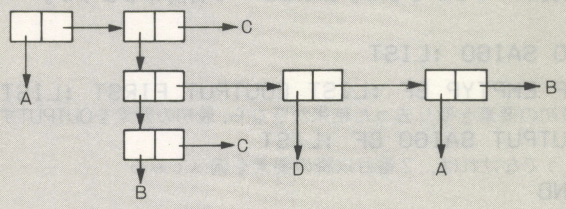


図2




```
TO JYO1 :N :M
```
JOY1は2つのインプットをとる
```
LOCAL "W
```
値が入るエリアとして"Wをとる
```
MAKE "W 1
```
"Wの値を1にする
```
REPEAT :M [MAKE "W :W * :N]
```
:Wに:Nをかけたものを"Wの値とする作業をM回繰り返す
```
OUTPUT :W
```
:Wの値を返す
```
END
```

余談だけど，OUTPUT　:WのところをPRINT　:Wにす

る人がよく見うけられるけど，そいつはまずい。NのM乗を求めたら必ず出力してしまうプロシジャは使いようがない。だから汎用性を保つために必ずOUTPUTし，PRINTするなら，

```
PRINT JYO1 3 5
```

のように，呼び出す側でやろう。

で，話を戻して，このようなプログラムは自然でわかりやすいし，別に間違いがあるわけじゃない。ここで別の考え方――それはきっと新しい万能性を与えてくれる。つまり再帰的――を導入すると次のように書ける。

```
TO JYO2 :N :M

IF :M = 0 [OUTPUT 1]
 n⁰は1である
OUTPUT :N * JYO2 :N :M - 1
 そうでなければn*n^(m-1)である
END
```

少しトレースしてみよう。

3の2乗を計算する

3の2乗は，3に3の1乗をかけたものである

3の1乗は，3に3の0乗をかけたものである

3の0乗は，1である

よって，3の1乗は3，3の2乗は9である

文章で説明するより慣れるほうが早い。次はリストを対象にしてみよう。リスト操作は再帰を使わなければできないものが多い。LISTはプリミティブだが，FIRST，BUTFIRST(BF)などを使って定義することができる。SAIGOって名前にしようか。

```
TO SAIGO :LIST
IF EMPTYP BF :LIST [OUTPUT FIRST :LIST]
最初の要素を取り去った結果が空なら，最初の要素をOUTPUTする
OUTPUT SAIGO BF :LIST
そうでなければ，2番目以降の要素を調べてみる
END
```

もしENPTYPでなく，EQUALP[ ]……とすると，ワードを引数として与えられなくなるから要注意。

次は，やはりプリミティブになっているCOUNTを，CNTという名で再定義してみよう。

```
TO CNT :LIST

IF EMPTYP :LIST [OUTPUT 0]
空なら0である
OUTPUT 1 + CNT BF :LIST
そうでなければ，最初に要素を取ったあと，CNTしたものに1を
加える
END
```

JYOもそうだったけど，このCNTも返ってきた値に対して計算をする。基本的なパターンとして覚えておくと楽だ。

では，〔A B C〕を引数として与えると〔C B A〕を返すような関数REVERSEを定義してみてくれ(解答は文末)。

また話はそれるけど，IF文の使い方がよくわからないという人が多い。IFはコマンドでもオペレーションでも使え，引数の数が不定(1か2)にもかかわらず( )で囲む必要がない，というのが原因のようだ。先ほどのCNTは次のようにも書ける。

```
TO CNT2 :LIST
OUTPUT IF EMPTYP :LIST [0] [1 + CNT BF :LIST]
END
```

また，このようにも使える。

```
?PR IF 1 = 1 ["SC] ["FA]
SC
```

つまり，IFコマンドの引数でなければコマンド，あればオペレーションである。

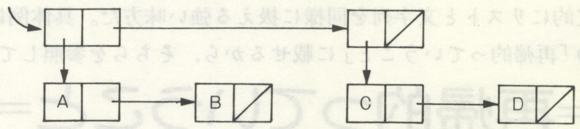
次に，再帰ってことからははずれるけどワードを少しいじってみよう。BASICのLEFT$, RIGHT$にあたるものがあると便利じゃないか？

リストAにLT$，**リストB**にRT$を示す。このように"Lに代入しないで，直接：WDを使ってもかまわない(**リストC**参照)。再帰的に作らないほうがよいだろう。

これらを使って文字の置き換えを行うSUBST――たとえばSUBST "ABCDEF 4 "Kが"ABCKEFとなるようなプリミティブを作ると，**リストD**のようになる。

═とりあえず今回は終わり═

2つ問題があったが，その解答は，



Logoが扱うデータの場合，ノードの右側にワードがくることはない，という点に注意してください。

2つ目の問題，REVERSEの定義は**リストE**のようになる。意味は自分で考えてみよう。

*　　*　　*

Logoはとてもフレンドリな言語で，やっているうちにいつの間にか覚えてしまう。文の説明を読むより，例をバンバン打ち込んで試行錯誤したほうがよくわかるってものだ。先日もBASIC

をなん日教えてもわからなかった女の子が，１日でタートル関係をみんなマスタしちゃったし，子供でも扱える。ただ，再帰だけはわかる人とわからない人の差が激しいようだね。言語が新しいから，なにがわからないのかよくわからない，読者のリアクションを待つ。アンケート葉書の片隅にでも書いてくれれば，とってもうれしい。

宿題

1） REVERSEを使って，リストの並びが対称かどうかを調べるオペレーションSYMMETRICPを作れ。

2） nの階乗を求める関数を作れ。

リストA

```
TO LT$ :WD :N
LOCAL "W
  LOCAL "L
MAKE "W "
MAKE "L :WD
REPEAT :N [MAKE "W WORD :W FIRST :L MAKE "L BF :L]
OUTPUT :W
END
```

リストB

```
TO RT$ :WD :N
LOCAL "W
LOCAL "L
MAKE "W "
MAKE "L :WD
REPEAT :N [MAKE "W WORD LAST :L :W MAKE "L BL :L]
OUTPUT :W
END
```

リストC

```
TO RT2$ :WD :N
LOCAL "W
MAKE "W "
REPEAT :N [MAKE "W WORD LAST :WD :W MAKE "WD BL :WD]
OUTPUT :W
END
```

リストD

```
TO SUBST :WD :N :AL
OUTPUT ( WORD LT$ :WD :N - 1 :AL RT$ :WD ( COUNT :WD ) - :N )
END
```

リストE

```
TO REVERSE :E
OUTPUT IF EMPTYP :E [:E] [SENTENCE REVERSE BF :E FIRST :E]
END
```

©1984 Thabio La Baugh

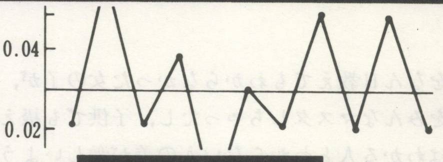
●統計のすべてがわかる通信講座！

統計解析 基礎力 応用力 養成講座

★統計の資格と実力！統計官（主事）資格が取れる！

文部省認定 社会通信教育

●唯一の文部省認定/行政管理庁指定/通信講座
●執筆・指導＝上智大教授・理博　斎藤金一郎
前東京大教授・理博　奥野　忠一
千葉大教授・理博　浅井　晃
慶応大講師・工博　芳賀　敏郎



●統計資料無料呈 ハガキか電話で直接下記までどうぞ！

★講座の特色★

- ●統計学を初めて学ぶ人、さらに理解を深めたい人に、今こそ、統計を自分のものにするタイムリーな講座。
- ●膨大な理論と広範な応用領域をもつ現代統計学の集大成決定版。
- ●統計の基礎から応用まで、豊富な実例によるシステム教材で短期指導。
- ●提出レポートに個別添削指導を実施
- ●修了者に行政管理庁指定《統計官・統計主事》資格証書を交付。

★主要内容★

第１単元＝統計とは何か
第２単元＝集団構造の記述
第３単元＝母集団と標本
第４単元＝推定と検定
第５単元＝回帰と相関
第６単元＝統計調査
第７単元＝標本調査法
第８単元＝品質管理と実験計画

㈶実務教育研究所　公開講座統計部
160 東京都新宿区大京町４の380
☎東京03(357)8153

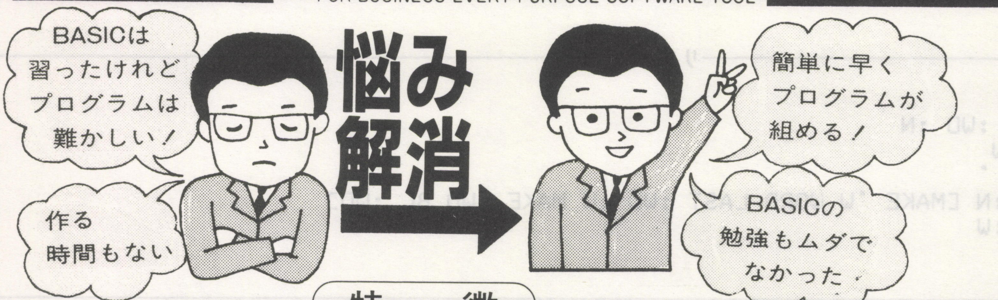
好評発売中

あなたも今日から プログラマー!!

BEST ベスト

FOR BUSINESS EVERY PURPOSE SOFTWARE TOOL

ビジネスユースの為の 簡易ソフトウェアツール!!



適応機種

- ●富士通FM-7
- ●富士通FM-8
- ●富士通FM-11
- ●NEC PC-8800
- ●NEC PC-9800

特　徴

1. 難しいプログラムをパーツ化！
   - ●難しいプログラムの部分をパーツ化してフロッピーに格納してありますからお客様は自由に取り出して、つなぎあわせ、必要な変数やデータを書き込むだけで、本格的なビジネスプログラムが作成できます。
   - ●パッケージソフトや、汎用ソフトと違い、内容は全くのハンドメイドですから利用者の思想が充分盛り込め、拡張性があります。

提供媒体　●フロッピィディスク（5インチ・8インチ）

※上記に関するお問い合わせは企画部 福岡・柘植まで。

日本コンピュータスクール事業協同組合加盟

理工産業株式会社
太陽グループ　〒510　三重県四日市市九の城町４-２０
TEL〈0593〉５１－１６５１㈹

- ●四日市教室 〒510 三重県四日市市九の城町4-20 ☎(0593)51-1651
- ●津　教　室 〒514 三重県津市本町23-17 ☎(0592)25-6251
- ●松阪教室 〒515 三重県松阪市船江町785 ☎(0598)51-4312

UE URBAN

アーバン オリジナルソフト

FBCP/MVI.O

FM-7・8・11用DISK版 ¥19,800（※2ドライブ必要）

F-BASIC↔CP/M ファイル転送プログラム

- F-BASIC上で使用
- 150クラスタ300KまでBASICTEXT及びシーケンシャル・ランダムファイル（データ）全ファイルをF-BASICからCP/MディスクへCP/MからF-BASICディスクへ1クラスタ(2K)単位の高速転送。
- CP/Mのファイル NAMEのチェック、DISKのSETmissのチェック等ミスオペレーションによる 事故のない様に万全のチェックが施してあります。
- CP/MディスクからF-BASICディスクへの転送の際、その都度F-BASICディスクのFATを退避させていますので停電等万が一の事故にも対処できます。

スペシャルコマンドII (PUSH-PULL)

F－BASICのプログラム中でスタック処理が扱えるようになります。

■コマンド一覧

init	スタック領域の初期設定を行います。
push	数値をスタックへpush します。
pull	変数へスタックの数値をへpull します。
exchange	スタックの先頭の数値と変数の内容を交換します。
usr0	スタックの先頭の数値になるユーザ関数です。

スペシャルコマンドシリーズのプログラムは、予約語を増やすものではありません。したがって他のプログラムとの共存が可能です。

ソースリスト添付 FM-7・8用 DISK版 ¥7,800円

第二弾

相性占い

パソコン相性占い

FM-7 DISK 版¥9,800—

♥恋人同志の彼と彼女の相性は？

彼女と互いに質問に答えてください。顔色が変わるかもネ!!

- ○質問は3ステップ 最後は人相判断も
- ○判定結果はプリンターに漢字出力されます。（本体に漢字ロム不要）

FM-11 スーパーぷりんていんぐユーティリティ

FM-11のユーザーへお待たせしました!!

●5インチDISK版 ¥12,800

ついにあなたのプリンタで（ドットインパクトタイプ）漢字印字が実現します。

FM CP/M86のPRINTINGがさらにパワーアップ!! その上F-BASIC(86)で漢字をふくんだLISTの印字もできます。

なお、印字の際に行間・横倍角・文字間隔・縦書き指定可能!!

RUN85で動作するCP/M80のソフトも印字できます。

漢字プリンターをもっていない方の必需品

このソフトがあれば、あなたのプリンターも漢字プリンターに なってしまう。ものすごいソフト……

●使用可能なプリンター

富士通シリアルドットプリンター・EPSON MPシリーズ RP-80・FP-80・FP-80K・KP-3000

インフォライナーRITEMAN・GP-550E

（CP/M 86上のルーチンですのでスーパーカルク等も印字可能）

★全国有名マイコンショップにてお求めください。

尚、通信販売も受付けております。送料500円

御振込先：住友銀行広島支店 当座201250

ミニファクス TEL (082)246-4592

UE URBAN アーバン電子株式会社

〒730 広島市中区大手町1-1-23 ☎(082)246-0993(代)

埼玉・熊谷 パソコンショップ CARC〔カーク〕

パソコンの真価はソフトで決まる！

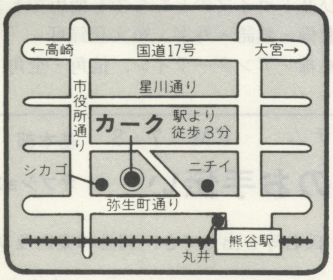
ソフト・専門図書充実のパソコン専門店

営業時間 AM10：00〜PM8：00

定休日 毎火曜日

取扱いメーカー

NEC
シャープ
富士通
東芝
カシオ
アップル
エプソン
ETC.



テープ版ワープロの決定版

好評発売中！

日本語簡易ワードプロセッサー 書あ句（かあく）

¥4,900

MODEL FUJITSU FM-⑦／FM-⑧ 要本体漢字ROM

適合プリンター EPSON RP-80. FP-80. MP-80TYPEIII

特徴

- メモリをフルに活用したテープ版では最高の機能と操作性
- スクリーンエディター方式で文章編集容易
- 一画面40桁×10行、400文字表示
- 漢字の呼出しはカナ2文字まで指定可
- 縦書き・横書き・改行間隔2種指定可
- この機能でこの低価格

本商品は全国マイコンショップ又は直接当社で（現金書留・送料無料）お求め下さい。 販売代理店／㈱日本ソフトバンク・誠光堂書籍㈱

コンピューター応用開発株式会社 TEL0485-25-4432

〒360 埼玉県熊谷市弥生町2－47

SBC

漢字顧客管理システム

新発売

FM-KOKYAKU

FM-7, 8

¥30,000
(標準フォーマットFD付)

32項目を自由にファイルフォーマット・変更もできる実用的な顧客管理です。

■仕様内容

▶最大項目数：32項目(自由に設定・変更可能)
▶漢字・熟語：20,000語(人名・地名も含む)
▶最大登録件数：3840件(1件64バイトの場合)
▶検　　索：すべての項目に検索条件ができるほか，多重検索も可能です。
▶検索条件：比較区分(＝・＞＝・＜＝)
全体区分(AND・OR)
▶印　　刷：一覧表(4種類)
ダイレクトメール(4種類)
印刷書式設定・変更が自由
データの分類処理・順番設定可能
▶プリンタ：シリアルドットプリンタ・熱転写プリンタ

FM−KOKYAKUは、項目桁数・データの型を自由に設定できますので顧客管理の他図書管理・その他登録・検索・印刷の作業にご使用いただけます。又、標準フォーマットFD付ですのでお買い上げいただいたその時からすぐお使いいただけます。

話題のヒットソフト好評発売中

日本語ワードプロセッサ　PC 8801　PC 8801 MK-Ⅱ
プリンターPC-8822・NK3618・RP-80・FP-80

PC-SWP　¥39,800

▶漢字6000字読み、熟語14,000語と豊富な辞書をもち追加もできます。
▶漢字変換方式はカナもローマ字も入力でき、お好きな方をインプットできます。
▶罫線・アンダーライン・見出し文字・縮小文字・縦書き出力はもちろん240種の任意の文字、記号を作成できる外字登録機能もあります。

日本語ワードプロセッサ　FM−7

FM-JWP/7　¥20,000

▶対応プリンタ：シアルドットプリンタ
▶カナ・ローマ字入力，文節変換で20,000語の辞書装備
▶1ドライブ及び2ドライブのどちらでも使用できます。
▶罫線は実線・細線・アンダーライン，文字は倍角・全角・半角
▶豊富な編集機能で計算処理も可能です。

漢字版作表簡易言語　FM−7

FM-PLAIN/7　¥20,000

▶集計表・一覧表・見積書などが誰でも簡単に作成できます。
▶20,000語の辞書装備・1ドライブでも2ドライブでも可能。
▶四則演算・分類・検索・複写など豊富な機能を装備。
▶自動罫線入力機能により操作が簡単です。

日本語ワードプロセッサ　FM−7・8

FM-JWP/S　¥39,800

▶対応プリンタ：シアルドットプリンタ・ビジネスプリンタ
▶1ドライブ及び2ドライブのどちらでも使用できます。
▶20,000語辞書装備・熟語・外字・短文登録可
▶実線・細線・点線・アンダーライン，倍角・全角・半角

酒類小売業システム　PC・9801・8801

酒屋さんのお手伝い　¥250,000

▶酒類小売販売店にとって最も繁雑な業務の税務署への報告書作成及び指定の酒類受払帳の作成等を自動的に行います。
▶出力帳表としては，仕入先別仕入月報・容量別在庫月報・品目別在庫月報・酒類別在庫月報・酒類受払帳等があります。

檀家管理システム　FM−7・8

お寺さんのお手伝い　基本部 ¥150,000　オプション ¥50,000

▶1000軒の住所・氏名等を出力でき，回忌の案内及び明細，宛名印刷ができます。
▶お布施，寄付金，年会費等の実績管理が出来ます。
▶過去台帳の登録，檀信徒の登録，弊社にて承ります。(有償)

SOFTWARE & BUSINESS・COMPUTER CO., LTD.

株式会社エス・ビー・シー

FM-KOKYAKUは実用的な顧客管理プログラムです。

・漢字の入力が簡単です。

日本語ワードプロセッサFM-JWP/7 と同様、カナ/ローマ字漢字変換により、漢字の入力がとても簡単です。

・実用性があります。

あらゆる業種に幅広くご使用いただけるように、データ入力画面、ファイルフォーマットが任意に設定できます。

・印刷書式の設定が自由にできます。

3種類のタックシールに宛名書印刷ができるほか、一覧表は任意の項目を任意の順で自由に印刷することができます。

・大量の情報処理が可能です。

一枚のフロッピィディスクに最大3,840件ものデータを登録でき、しかも高速でアクセスできます。

・すべての項目に検索条件が設定できます。

すべての項目に検索条件が設定できるほか多重検索も可能です。

会社名	担当者名	電話番号	〒	住所
富士通株式会社	富士通夫	03-502-0161	105	東京都港区虎ノ門2-3-13
三和ビジネス株式会社	三和太郎	045-641-6401		市中区山下町75番地
岡田商事株式会社	岡田一郎	03-450-8092	108	都品川区北品川3-7-28
渋谷電気商会	山本和雄	03-744-8910	150	東京都渋谷区道玄坂1-2-6
内田電子工業株式会社	内田良一	03-541-3310	100	東京都千代田区内神田3-3-1
日本システム株式会社	日本一郎	045-201-8917	235	横浜市南区上大岡2-6-1
コンピュータジャパン株式会社	佐藤明弘	011-755-1111	060	札幌市中央区北二条東３丁目
株式会社三和エンジニアリング	高島利雄	0493-673-8904	355	埼玉県浦和市道場９５番地

〒105
東京都港区虎ノ門2-3-13
富士通株式会社
富士通夫 殿

〒231
横浜市中区山下町75番地
三和ビジネス株式会社
三和太郎 殿

〒108
東京都品川区北品川3-7-28
岡田商事株式会社
岡田一郎 殿

〒150
東京都渋谷区道玄坂1-2-6
渋谷電気商会
山本和雄 殿

〒100
東京都千代田区内神田3-3-1
内田電子工業株式会社
一 殿

〒235
横浜市南区上大岡2-6-1
日本システム株式会社
日本一郎 殿

FM7・8・11用

財務会計システム

PC−9801・9801F
MZシリーズ・
FM−7・8・11用

FM−ZAIMU

- 1枚のミニフロッピーデスクィに最大10,000件もの仕訳が入力保存できます。さらにミニフロッピーディスクを複数にすることによって、最大14,000件の仕訳伝票処理に対応できます。
- 仕訳伝票入力時の摘要入力、勘定科目名の修正等、漢字の入力はカナ漢字変換方式が使用でき、漢字の取り扱いはとても簡単です。
- カナ漢字変換による摘要入力の他に固定摘要のコード入力や更新後の仕訳伝票の修正など随所に豊富な機能を取り入れ、高い操作性を実現しました。

三和ビジネス(株)製　基本部 ¥150,000
オプション ¥50,000

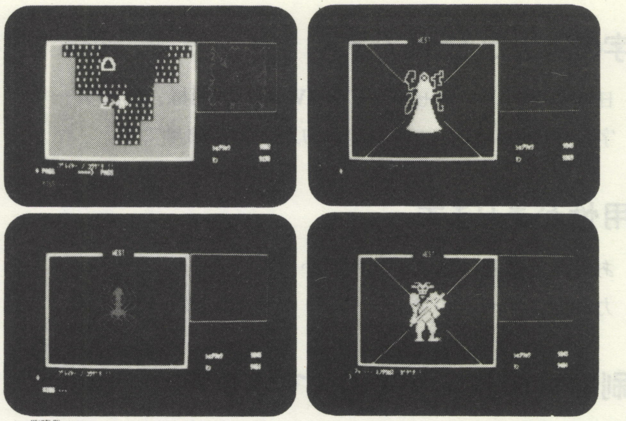
PCA高級財務会計システム

- あなたの会社に合わせた科目登録や補助簿登録ができ、合計金額1,000億円未満の会計処理ができます。
- 主票帳票はオフコン並の専用用紙が用意されておりますので、銀行や税務署等にそのまま提出できます。
- 画面は漢字の大文字で表示され、非常に見やすくなっています。
- 日常良く使われる主要48科目は、ワンタッチ入力方式になっています。
- 入出金データーの入力は自動仕訳方式にする等、使いやすく、正確で、しかも主要部分は機械語処理により、トータル・パフォーマンスを追求しています。
- 処理範囲によりA.B.Cと3段階のシステムが用意され、必要なレベルを選択できます。
- 会計専門家数名のノウハウを結集したシステムです。

ピーシーエー(株)製
¥98,000 ¥148,000 ¥198,000

横浜市中区本町5−49(丸忠センタービル3F 〒231)
☎045-201-8480(代)
ファクシミリー 045−201−8386

ソフトのグレードが、マニ



画面写真PC-8001用（PC-8801、FM7のグラフィックスに絶対の自身あり）

ダンジョン（地下迷宮）

プロデューサー　シブサワ コウ

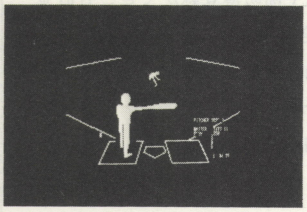
今アメリカで最も人気があり、ゲームの主流であるロールプレイングゲーム。これを思考ゲームの殿堂光栄マイコンシステムが創ると、どうなるか。もちろんテープ版の極限もきわめてありますぞ。さあダンジョンへ旅立つ貴方、光栄のファンタジーロールプレイングゲームでこのだいごみを十二分にお楽しみあれ！

ダンジョンは、日本のコンピュータゲーマー達にロールプレイングゲームの存在を身近に感じさせてくれる格好のソフトだ。ゲーム進行も比較的シンプルだし、ストーリーも日本人にとって理解しやすい。ロールプレイングゲーム入門に最適。（ログインより）

●PC-8001、8801、FM7　テープ版**5,200円**（ユーザーステープ付）

思考ゲームを国内に定着させた光栄が
新春に放つ史上最高のアドベンチャーゲーム

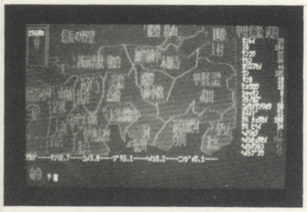
「ルイン」　4月発売予定



ペナントレース

●テープ4,500円●5′ディスク6,800円
●機種PC-8001/8801/FM7/11

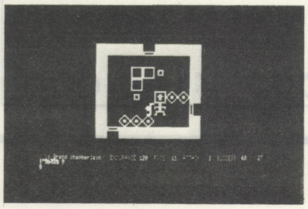
秋葉原サトー無線様、連続8週1位獲得
その他多数ショップ様にてトップランキング続出！



信長の野望

●テープ4,500円●5′ディスク6,800円（PC-9801用8′ディスク7,800円）●機種PC-8001/8801/9801/9801F/FM7/8/11 X-1

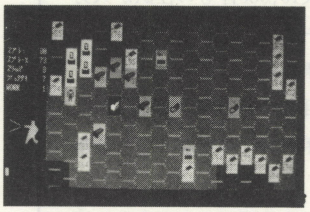
〝もう目が血走っちゃいそう！ 信長につき合って二晩徹夜してしまったと言ったら読者は笑うだろうか。しかしこれホントの話〟。「ポプコム」より



クフ王の秘密

●テープ4,500円●機種PC-8001/8801/FM7

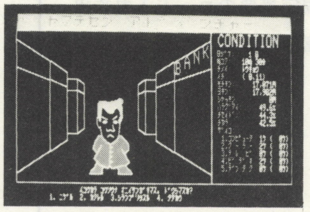
国内第一号のロールプレイングゲーム
〝パソコンゲーム界の天才シブサワコウ作〟



ホイホイ

●テープ3,500円 PC-9801用ディスクパック2本組7,800円
●機種PC-8801/FM7/X-1C・X-1

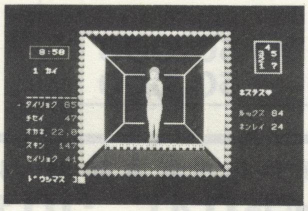
〝爆笑度100%のパロディ精神に脱帽！手法はあくまで正統的なウォーゲームを踏え、コミカルゲームに仕立てあげたこの手腕。さすが光栄さんエライ。文句なしのお買得の秀作といえよう!!〟「パソコンゲームランキングブック」より



アドベンチャーパック

●テープ3,800円●機種PC-8001/8801/FM7/8

〝山手線アドベンチャー〟〝コンピュータジャック〟二本のグラフィックアドベンチャーゲームが入った超お買得品。



団地妻の誘惑

●テープ4,800円 ●5′ディスク6,800円（PC-9801用8′ディスク7,800円）●機種PC-8801/9801/9801F/FM7/8（ディスク版FM7/8/11）

〝ひょっとしたらこのゲームはノルマに追われる我々現代人の生活への作者の強烈な文明批判ではないかという気さえしてくる。ゲームとしてはグラフィックや様々なアイディアで充分楽しめる上出来のソフト。〟「カーソル」より

アのニーズを越えた。

FM-7遂に完成!

調査ゲーム360本中
●シミュレーション部門最高得点獲得
●ランキング第一位獲得

「ホイホイ」

スポーツゲームの大傑作!!

「ペナントレース」

(2人プレイ用)

すごいヘクス画面スクロール
★新作★アルデンヌの戦い ●シミュレーションウォーゲーム ●PC-8801 テープ¥3,800

N88ベーシックコンパイラ

国内初、遂に登場!速さ100倍 テープ版 ¥10.000

おみせできなくて
ゴメンナサイ!

ナイトライフ
●テープ4,500円●5′ディスク7,800円
(PC-9801用8′ディスク8,800円)●機種
PC-6001MkII/6601/8001/8001MkII/8801/9801/9801F/FM7/8/11/MZ2200/5500/X-1
テレビ、新聞・雑誌にと、これ程話題に上ったソフトは、かつてなかった。世界に類を見ないアダルトの世界。

日本語ワードプロセッサー PC-8001

「い・ろ・は」

PC-8001で漢字!
★ソフトバンク日本語ワープロ出荷ベース第5位
PC-8801・PC-8001 MKII(N-BASIC可) 29,800円

●マシン語による高速処理●漢字ROM不要●漢字プリンタ不要●本体改造不要●熟語登録1万語●拡大文字・アンダーライン・縦書き機能その他豊富
▶使用可能プリンター/PC-8023C/PC-8821/PC-8822/MP-80II/MP-80III/RP-80/FP-80/FP-80K/GP-500E/GP-250F

ビジネスマン必携!! ビジネスシミュレーションゲーム

「トップマネジメント」

●5インチ、8インチディスク版●カナ版16,800円●漢字版19,800円
(PC-9801F用のみ)

製造、販売、財務等、会社経営の全てをシミュレートした本格的マネジメントゲームです。経営者として日々決断すべき事柄の大部分を網羅してありますので、自己啓発用、企業の社員教育用に是非御活用下さい。

〈ディスク版〉
PC-8001、PC-8001MkII、PC-8801、PC-8801MkII、PC-9801、PC-9801F
(PC-9801F漢字版、FM7、FM8、FM11、開発中)

汎用漢字顧客管理システム ★日経パソコンベストヒット商品 ランキング第5位

「商売繁盛」

V2・0 FM-7/8 29,800円

●1ディスケットあたり600件の顧客情報●仮名漢字変換方式により簡単に漢字登録●業種に合せて、ユーザー独自の項目設定可能●メニュー表示により、選択できるので操作簡単、その他豊富な機能盛りだくさん
▶使用可能プリンター/MB-27403/MB-27404/MB-27405/MP-80II/MP-80III/RP-80/FP-80/FP-80K/GP-250F/GP-550E/MP-80K/PC-8822

光栄マイコンシステム★

本社:〒326 足利市通1丁目2677 ☎0284-41-5911(代)
支社:〒233 横浜市港北区日吉本町1876 ☎044-61-6861(代)

■お求めは全国のマイコンショップ又は本社宛現金書留にてお送り下さい。(送料サービス)

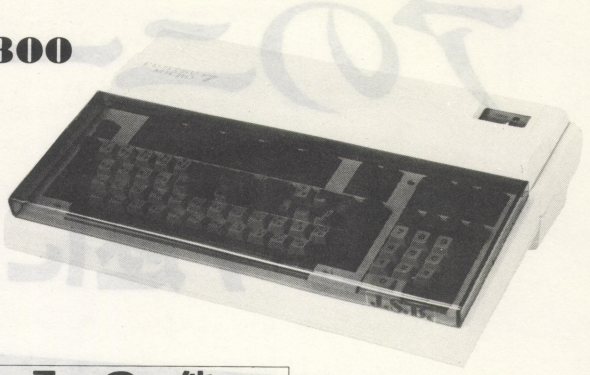
資料請求券 FM 4

マイコン買ったらキーボードも買おう！

キーボードカバー

(各)¥3,800

SOFTOP



富士通パソコン用

- ★FM-7
- ★FM-8
- ★FM-11

NECパソコン用

- ★PC-9801
- ☆★PC-9801E
- ☆★PC-9801F
- ★PC-8801
- ★PC-8801MKII
- ★PC-8001
- ★PC-8001MKII
- ★PC-6601
- ★PC-6001
- ☆★PC-6001MKII

シャープパソコン用

- ☆★X-1
- ☆★X-1D
- ☆★X-1C
- ★MZ-2200
- ★MZ-700

MSX用

- ☆★ ナショナル CF-2000
- ☆★ ソニー HIT BIT HB-55
- ★ 東芝 PASOPIA IQ HX-10D, 10S
- ☆★ サンヨー WAVY-10
- ☆★ 富士通 FM-X

その他

- ★ 東芝 PASOPIA-7
- ☆ セガ SC-3000
- ★ 日立 ベーシックマスター L3 (2M, M5)
- ★アップルII, IIe

カラー	
	☆：透明
	★：ブラウン・スモーク
材質：高級アクリル	

お求めの際は、透明、ブラウン・スモークのいずれかの色を御指定下さい。

- ■：キーボードのまわりまでかぶさるタイプ
- ■：キーボードの一部がかぶさるタイプ
- ■：キーボードの上にのせるタイプ

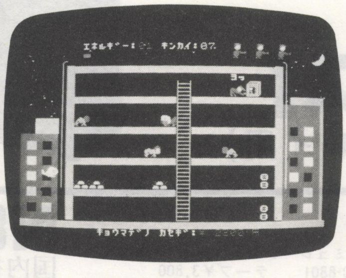
マイコンカバー（布製）
MZ-2000、MZ-80B用も取扱っています。
●カラー：エンジ・紺
¥3,000

SOFTOPのゲームソフト

あぶく丸の冒険PART 1

(X-1シリーズ) IPL G-RM 型番 SX-001 ¥3,500 JOY

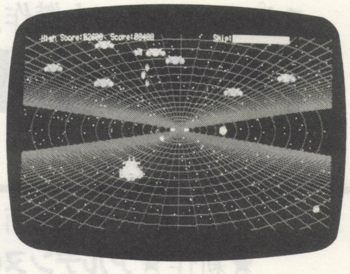
次々とおそってくるタコのペトリゴンに立ち向かう、あぶく丸。落ちてくる岩のクッションを使ってペトリゴンをたおせるか。



ザ・ガードマン

(X-1シリーズ) IPL G-RM 型番 SX-005 ¥3,500 JOY

きみは9個のキー操作をどこまでこなせるかな。金塊を運ぶきみを待ちうける者は……



ブラックホール

(X-1シリーズ) IPL G-RM 型番 SX-006 ¥3,300 JOY

きみのすごうでに、エイリアンもおもわずむしゃぶるい。ブラックホールの向うにあるものは。

スーパーボウル

(X-1シリーズ) IPL G-RM 型番 SX-003
(FM-7) 型番 SF-001 ¥3,500

ハイ・ゲーム賞はだれの手に…

あぶく丸の冒険PART 2

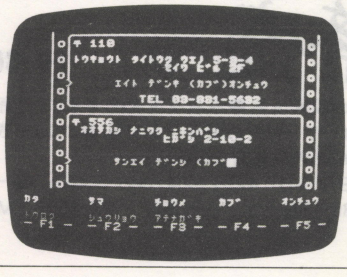
(X-1シリーズ) IPL G-RM 型番 SX-002 ¥3,500 JOY

1面クリアーするごとに向きが変わる。寅返りしたあぶく丸を君は見れるか。

エイリアン・メイズ

(X-1シリーズ) IPL G-RM 型番 SX-004 ¥3,300

迷路の中に3個のコイン。全部拾ったら出口へ急げ!! おっと、エイリアンがまちかまえてる。



宛名書き専用ソフト（住所録・電話帳）

ゆうびん屋さん

(X-1シリーズ) 型番 SX-501 ・(FM-7) 型番 SF-501 ¥3,800

手書きの苦労が ふっとんだ!!

「ゆうびん屋さん」は機能を限定して、めんどうな操作や設定を除いた結果、はじめての方でも簡単に使っていただけるようになりました。

別売品

マイコン用 (型番 TC-10)

宛名タックシール

- ●4インチ幅トラクタフィールド用紙
- ●1ケース・1ロール500枚入×2コ入
- ●シールサイズ(タテ)38㎜×(ヨコ)86㎜

¥3,200

■お求めは……（商品送料はサービスいたします）
全国パソコンショップまたは直接右記へお申し込みください。
- ●現金書留でご注文の場合は商品名、住所、氏名、電話番号をはっきり書いてお送りください。
- ●銀行振込みでご注文の場合は商品名、住所、氏名、電話番号をハガキまたは電話でご連絡ください。

企画・製作 ソフトップ

発売元

エイト電気株式会社
〒110 東京都台東区上野5-3-4 ☎03-831-5632(代)
振込銀行：北陸銀行 上野支店 (普)No.4032110

三栄電子株式会社
〒556 大阪市浪速区日本橋東2-10-2 ☎06-643-3833(代)
振込銀行：三和銀行 恵美須支店 (普)No.189879

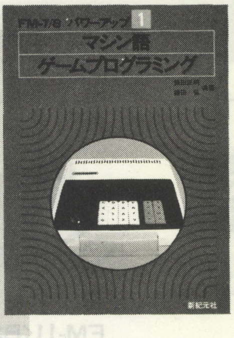
FM-7/8パワーアップ 1

マシン語ゲームプログラミング

鎌田正明/鎌田 弘 共著

既刊

○B5判 224頁
○定価2,200円



アセンブラから ゲームプログラミングへ

本書は，アセンブラの入門に始まり，ゲームプログラミングを通して，サブCPUの使い方を理解してもらうことを目的としているものです。

●主な内容

1章 アセンブラ入門
アセンブラの基本／簡単なアセンブラプログラム

2章 入出力システム
BIOSの利用／ディスプレイサブシステム

3章 サブCPUを使った高速ゲーム
いろいろなマシン語ルーチン／テニスゲーム／ブロックくずしゲーム／コックローチゲーム／バックグラウンドミュージックの使用

4章 メインとサブCPUを使った高速ゲーム
アストロウォーズゲーム／ギャラクシーウォーズゲーム

付録 用語解説／ニモニック表 他

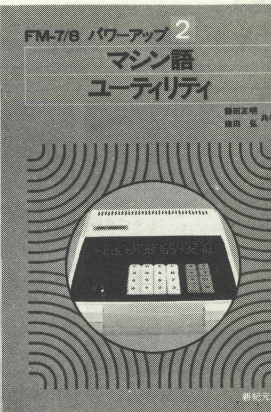
FM-7/8パワーアップ 2

マシン語ユーティリティ

鎌田正明/鎌田 弘 共著

既刊

○B5判 256頁
○定価2,500円



ユーティリティが充実してくると，
ハードなマシンが
ソフトなツールになる!!

FM-7/8の機能をもっと活かしたいと考えているみなさんのために，いろいろなユーティリティソフトをつくりました。これで周辺機器を含めたFM-7/8のシステムが，さらにみなさんのよきパートナーとなるはずです。

●主な内容

1章 マシン語エディタ
2章 プリンタやフロッピーディスクの活用ソフト
3章 クイックソートと応用
4章 F-DOS上のグラフィック機能強化
5章 音符入力エディタと演奏
付録 F-BASICの追加命令

◎「マシン語ユーティリティ」で掲載したプログラムを，カセット版で同時発売!!

マイコン宇宙百科事典
ユーティリティシリーズ(3)

マシン語ユーティリティ（カセット版）

●「マシン語ユーティリティ」の本に掲載されたプログラムのほとんどを収録しましたが，一部削除したものもあります。

○定価5,800円（〒240円）

マイクログループ MICRO GROUP

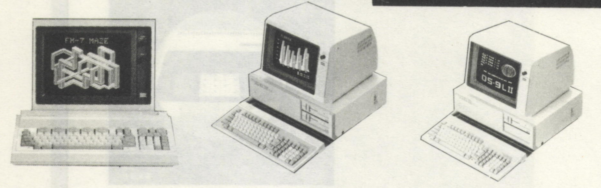
株式会社新紀元社 東京都新宿区新宿4-1-9新宿ユースビル
電話03(350)0848 郵振東京1-27618

電話で深めようマイコンの世界

106 106 106 106 106 無料 106 106 106 106 106 無料 106 106 106 106 106

アコリットマイコン最新カタログ

富士通ＦＭシリーズ



FM-11〈BS〉　FM-11〈AD2〉

FM-7(MB25010)　¥126,000
48回分割　¥3,100　／　60回分割　¥2,600

FM-11〈AD2〉　¥298,000
48回分割　¥6,854　／　60回分割　¥5,840

FM-11〈BS〉　¥398,000
48回分割　¥9,154　／　60回分割　¥7,800

型名	品名	定価/円	48分割	60分割
MB 27312	グリーンディスプレイ	49,800→	1,200	1,000
MB 27302	グリーンディスプレイ	46,800→	1,200	900
MB 27304	グリーンディスプレイ	29,800→	700	600
MB 27331	カラーディスプレイ	188,000→	4,700	4,000
MB 27301	カラーディスプレイ	188,000→	4,700	4,000
MB 27307	カラーディスプレイ	155,000→	3,900	3,300
MB 27303	カラーディスプレイ	99,800→	2,500	2,100
MB 27305	カラーディスプレイ	64,800→	1,600	1,400
MB 22602	家庭用ＴＶアダプタ	13,500→	300	300
MB 27402	ビジネスプリンタ	350,000→	9,000	7,400
MB 26201	フロントインサータ	89,000→	2,200	1,800
MB 27406	シリアルプリンタ	89,000→	2,200	1,800
MB 27404	ドットプリンタ	89,000→	2,200	1,800
MB 27501	データレコーダ	12,800→	300	200
MB 22406	I/Fカード	15,000→	400	300
MB 26001	拡張ユニット	133,000→	3,300	2,800
MB 22204	I/Fモジュール	30,000→	700	600
MB 22203	計測制御I/F・M	61,000→	1,500	1,300
MB 22209	音声入出力M	98,000→	2,400	2,000
MB 22405	漢字ROMカード	35,000→	900	700
MB 22002	キャラクタセット	10,000→	300	200
MB 22003	漢字キャラクタセット	30,000→	700	600
MB 28111	漢字ROMカード	46,000→	1,100	900
SM 07217 F061	システムカセット	39,000→	1,000	800
MB-27651	ディスクユニット	498,000→	12,500	10,600
MB-22416	ディスクI/Fカード	14,000→	400	300
MB-22211	ディスクI/F・M	20,000→	500	400
MB-27603	フロッピディスク	440,000→	11,000	9,400

型名	品名	定価/円	48分割	60分割
MB 27513	フロッピディスク	350,000→	8,700	7,400
MB-22415	Fコントロールカード	30,000→	700	600
MB-22604	フロッピアダプタ	58,000→	1,400	1,200
MB-27061	M/Fディスク	313,000→	7,800	6,600
MB-27605	S/Fディスク	148,000→	3,700	3,100
MB-27606	増設M/Fディスク	90,000→	2,300	1,900
MB-27607	S/Fディスク	98,000→	2,400	2,000
MB-27608	M/Fディスク	69,000→	1,700	1,400
MB-27609	M/Fディスク	65,000→	1,600	1,300
MB-22603	M/Fアダプタ	17,000→	400	300
MB-22407	M/F I/Fカード	14,800→	400	300
MB-28021	Z-80カード	35,000→	900	700
MB-28041	8088カード	80,000→	2,000	1,700
MB-28031	Z-80カード	45,000→	1,100	900
MB-22605	バブルホルダ	89,000→	2,200	1,800
MB-22609	バブルアダプタ	13,000→	300	300
MB-22417	バブルI/Fカード	11,000→	300	200
MB-22608	バブルメモリガード	58,000→	1,500	1,200
FBM-C128GP	バブルカセット	37,000→	1,000	700
MB-22601	バブルホルダ	85,700→	2,100	1,700
FBM-43CP	バブルカセット	35,000→	1,000	700
MB-22401	Z-80カード	11,700→	300	200
MB-28011		50,000→	1,300	1,000
MB-28012	8088カード＋5″	99,800→	2,500	2,100
MB-28013	CPM/86®	50,000→	1,300	1,000
MB-28014		99,800→	2,500	2,100
MB-22206	8088拡張RAM・M	49,800→	1,200	1,000
MB-22414A	拡張RAMカード	58,000→	1,500	1,200
MB-22414B	拡張RAMカード	88,000→	2,200	1,800

おもしろコンピュータ

カシオ　楽がき　PV-2000
¥29,800→¥23,800

セガ　コンピュータビデオゲーム
SG-1000　¥15,000→¥12,000
SC-3000　¥29,800→¥23,800

任天堂　ファミリーコンピュータ
¥14,800→¥11,800

●コレクトコールでのお問い合せは御遠慮ください。
●この商品のみは1500円をたして現金書留にてお願い致します

お買得コーナー

■カシオ	EP-1100＋ソフト＝¥65,000
■NEC	EP-1100＋ソフト＝¥65,000
■ナショナル	●JR-100＋RFコンバーター＋ソフト3巻＝¥30,000 ●JR-200＋プログラム集(1冊)＋ソフト2巻＝¥40,000
■シャープ	MZ-731→¥83.000

■オーディオ・キーボード・パーソナル無線もお問い合わせください。

注目のMSX対応各社パソコン好評受付中！

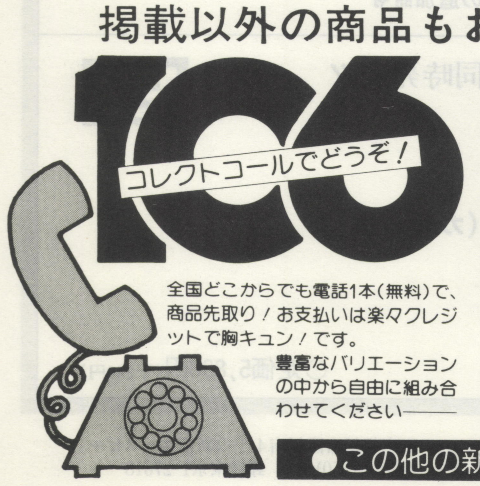
型名		品名	定価/円	15分割
日本楽器製造	YIS-303	本体	49,800→	3,347
	YIS-503	本体	64,800→	4,355
	SFG-01	ＦＭサウンドシンセサイザユニット	19,800→	1,330
	SMD-01	MIDIユニット	12,800→	860
	ZPA-01	プレイカードセット	12,800→	860
	YK-01	ミュージックキーボード	17,800→	1,196
	RF-01	家庭用カラーＴＶアダプタ	7,800→	530
	YRM-12	ＦＭ音色プログラム	7,800→	530
ゼネラル		PCT-50(キーボード付)	146,500→	9,298
富士通		FM-X	49,800→	3,347
東芝〝パソピアQ〟		HX-10D	65,800→	4,422
		HX-10S	55,800→	3,750
日立		MB-H1	62,800→	4,220
		MB-H1E	54,800→	3,683
三菱		ML-8000	59,800→	4,019
ソニー		〝HIT-BIT〟HB-55	54,800→	3,683
松下電器		CF2000	54,800→	3,683
三洋電機		MPC-10	74,800→	5,027
キャノン		V-10	54,800→	3,683
ビクター		HC-6	64,800→	4,355

Oh-FM 愛読者特別セール

テレビ		標準価格	48回均等	60回均等
ナショナル	TH21-H55GR	¥219,000	¥5,037	¥4,292
〃	TH19-L55GR	¥188,000	¥4,324	¥3,684
〃	TH15-M55GR	¥139,000	¥3,197	¥2,724
〃	TH14-NV1	¥208,000	¥4,605	¥3,924
ソニー	KV-20XR2	¥199,800	¥4,481	¥3,818
〃	KV-18XR2	¥179,800	¥4,032	¥3,436
ビクター	AV-260	¥290,000	¥6,527	¥5,562
〃	AV-200	¥210,000	¥4,709	¥4,013
東　芝	21K691	¥215,000	¥4,821	¥4,108
〃	15K31	¥129,800	¥2,910	¥2,480
シャープ	C-1 19型	¥145,000	¥3,252	¥2,771

ビデオ		標準価格	48回均等	60回均等
ナショナル	NV-630	¥169,800	¥3,613	¥3,203
〃	NV-330	¥159,800	¥3,170	¥3,015
ナショナル	NV-800	¥289,800	¥6,665	¥5,680
〃	NV-850HD	¥239,800	¥5,238	¥4,700
ソニー	SL-F5	¥169,800	¥3,808	¥3,245
〃	SL-F3	¥145,000	¥3,169	¥2,771
〃	SL-HF77	¥299,000	¥6,877	¥5,860
〃	SL-HF66	¥249,800	¥5,745	¥4,896
ビクター	HR-D225	¥195,000	¥4,373	¥3,726
〃	HR-D120	¥148,000	¥3,310	¥2,842
日　立	VT-16	¥173,000	¥3,880	¥3,306

掲載以外の商品もお問い合せ下さい。



全国どこからでも電話1本(無料)で、商品先取り！お支払いは楽々クレジットで胸キュン！です。

豊富なバリエーションの中から自由に組み合わせてください――

Acolyte
(アコリット)
〒110 東京都台東区北上野2丁目12番12号
営業時間／午前11時～午後7時　年中無休
☎03(844)3361

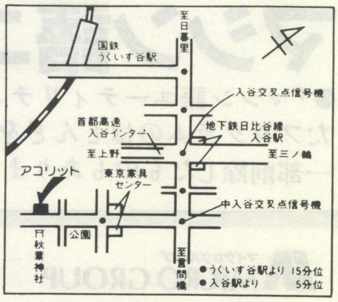
●他にオーディオ、ビデオ、パーソナル無線も取り扱っています。クレジットは1～60回までＯＫです。
●初回金のみ、端数金額処理のため、上記金額より少し多くなります。
●契約後の取消は規定の手数料を申し受けます。

●この他の新製品の御予約も受付けます。

お支払いは楽な長期クレジット

48回/60回

●現金の場合はお問い合わせ下さい。クレジットは3,000円以上の組み合わせに願います。



MIA
COMPUTER BOOKS & PACKAGE SOFTWARE

TAPE & BOOK

MIAテクニカル・ライブラリ 1
FM-7 高速ゲームの制作

A5判　定価3500円(カセット・テープ付)

「FM-7で面白い高速リアルタイム・ゲームを作りたい」という人のために、6809アセンブリ言語の基本、サブシステムの使い方、特殊な画面処理(仮想VRAM)・重ね合わせの処理等、プログラミングに役立つ実践テクニックをご紹介。(FM-7、リアルタイム・ゲーム付)

- ●STEP1/アセンブリ言語について…6809アセンブリ言語の特徴
- ●STEP2/サブシステムの使い方…共有エリア/メインテナンス・コマンド/コマンド・ハンドラ/サブシステムへのプログラム転送
- ●STEP3/ゲーム作りに有用なサブルーチン群…カラーパレットの設定/PSGコントロール
- ●STEP4/画面処理…グラフィック・パターンの表示、移動、重ね合わせ/仮想VRAM
- ●STEP5/ゲーム・プログラム…(真夜中のタイガー戦車)

パソコンゲーム傑作選集 4 ハドソン編 FM-7

1500円(〒240円)

雨の日は大忙し／ヒヨコファイター／シーボンバー／キャノンボール／Justice Knight／カラーボール／キラーステーション／ミスターバタフライ

ゲームフリークのための
アップル・ソフトウェアブック
Book For People Who Loves Apple.

APPLE FARM

A4判 3500円(郵送無料)

● THE STORY OF APPLE II

APPLE IIの歴史とその魅力を探る

なぜ、今、APPLEなのか、その人気の秘密は？　APPLEにとりつかれた マニアが語るAPPLEの魅力。

● THE WORLD OF FANTASY ROLE PLAYING GAME

WIZARDLY, ULTIMA 徹底研究

GAME SOFTの最高峰WIZARDLY, と、ULTIMAをあらゆる方向から徹底的に解説。

● THE CLASSIC GAMES

APPLE SOFT 名作選

数多くのAPPLE SOFTの中から名作と思われる30本をピックアップして解説。ラスター・ブラスター、ロードランナー、クライシス・マウンテン、ギャラクティック・サガetc.

● THE SOFTWARE 1980, 81

1980年81年に出たAPPLE SOFTを紹介

● THE SOFTWARE 1982

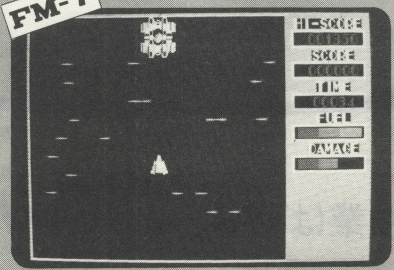
● THE SOFTWARE 1983

● FOUR GREAT PROGRAMMERS.

スタープログラマー、ビル・バッジ、ナシャ・ケベック、ダグ・カールストン、ケン・ウィリアムズの横顔とそのディスコグラフィー

キャプテン・スミス

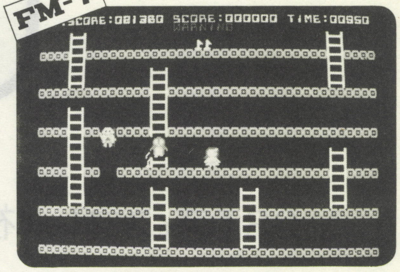
FM-7　2800円(〒240円)



ロングセンサーを頼りに未知なる敵を探し求めてワープ(時空間移動)。レーダー、ビーム砲…を駆使して敵船団と戦う。FMスペースアクション・ゲームの最高傑作。

おてんばベッキーの大冒険

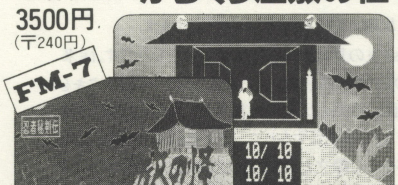
FM-7　2800円(〒240円)



ベッキーちゃんを追ってくるエイリアンたちを退治する方法はただ1つ。穴を掘りながら階段を昇ったり降りたり大忙しのベッキーちゃん…冒険物語のはじまり、はじまり。

忍者秘剣伝 からくり屋敷の怪

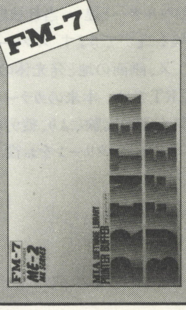
FM-7　3500円(〒240円)



秘剣「ムラクモ」のありかが記されている巻物を盗みだすために不気味な屋敷に忍び込んだ君は、侍、忍犬、毒ヘビ…を倒しながら迷路を進む。FMロール・プレイング・ゲーム。

プリンタ・バッファ

FM-7　ME-2 TAPE　4800円(〒240円)



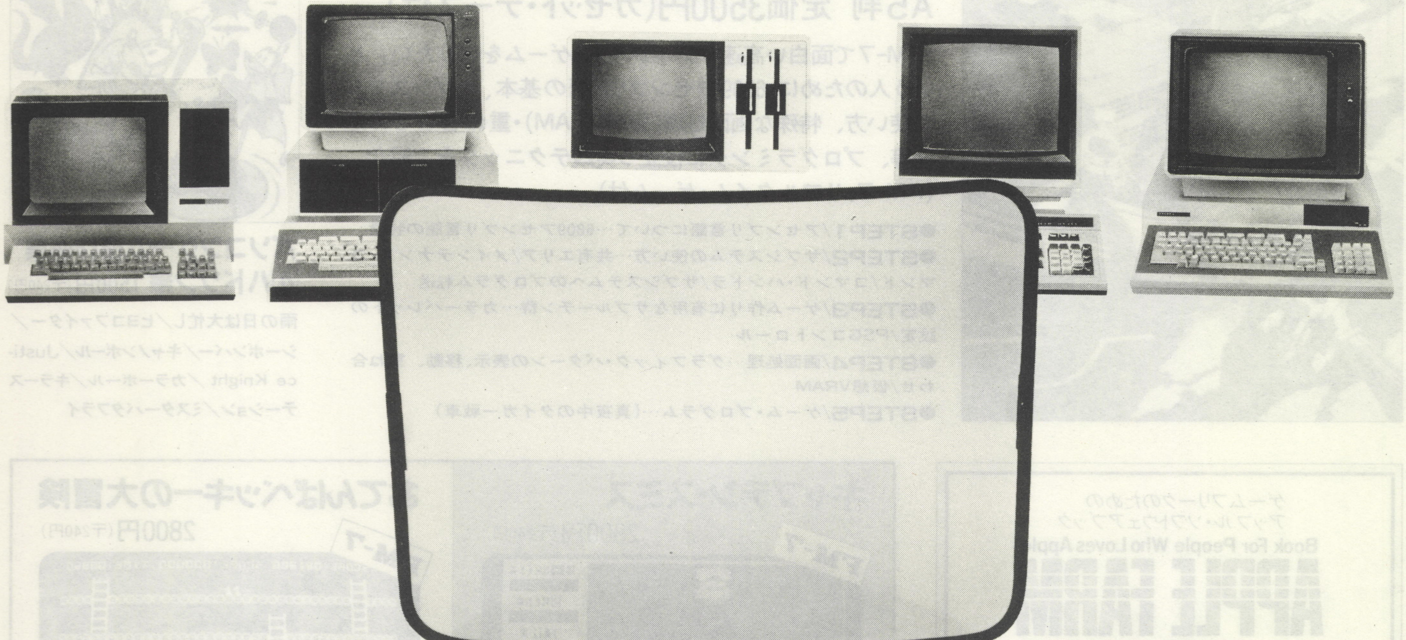
これまでハードウェアで行なっていたプリンタ・バッファ機能をソフトウェアで実現。割り込み処理でプリンタへの出力と同時に、プログラムのエディットや実行などの作業が可能に。BASIC命令で手軽に使用でき、プリンタに出力する命令はすべて(LLIST, HARDC, LPRINTなど)サポート。

(株)エム・アイ・エー
MIA MICRO INFORMATION ASSOCIATES
■お求めは最寄りのマイコン・ショップへ。あるいは郵送料(〒円)を添え現金書留で直接、当社へお申し込みください。
〒102 東京都千代田区紀尾井町3-20 鶴屋ビル ☎(03)265-2461

眼を大切に、だから…

E・Pスクリーン

Eye Protection Screen



CRTディスプレイ視作業は　なぜ眼が疲れるのか？

(1)私達は、家庭用TVを、慢然と見ていますが、CRTディスプレイ作業では、判読し難い文字を**注視**しなければなりません。それに加えて、作業時の精神的緊張も、疲労の原因と考えられます。

(2)CRT(ブラウン管)画面には、特有のチラツキやマタタキがあり、これらは目を疲れさせるばかりか、中枢神経へ影響を及ぼすともいわれています。

(3)CRT画面の文字は、明るく発光している点の集まりです。至近距離(30〜70cm)で長時間見つめることは、目に負担をかけることになります、。

(4)CRT画面に、外光や天井照明が写り込むと、反射が眩しく、煩わしいものになります。

E.Pスクリーンは、アーゴノミクス(ergonomics)※の見地から研究開発されたものです。従来の透明着色フィルターと違い、当社独自開発のD-M方式を採ることにより、チラツキやマタタキを効果的に押えることができます。スクリーン表面は、外光の反射を緩和するノングレア処理が施されています。又、画面の地と発光体のコントラストを適正にするので、文字をより見やすくします。特にカラーCRTでは、本来のカラーをより鮮やかに際立たせることが出来ます。EPスクリーンは、産業医科大学の実験により、疲労緩和に効果のあることが実証されております。職場の環境整備の一助にEPスクリーンをお役立て下さい。

	CRTの型名	価格		CRTの型名	価格
IBM	5251, 5255	14,500	SORD	DUS-12G	13,800
	3278	14,500		♯223	13,800
	6580	14,000		♯243	13,800
富士通	6262R	14,500		♯243C	15,000
	6262K	16,000	XEROX	14C	16,000
	6652A	14,500	東芝	JW-10	13,800
	9526R	14,500		JW-7	14,500
NEC	PC-8048K	15,000	日立	560-20	14,500
	PC-8049K	15,000		560-20-C	16,000
	PC-8050K	13,800		2171	16,000
	PC-8053	16,000	DEC	VT-100	13,800
	PC-8853	16,000	WANG	DIS-140	13,800
SORD	DUH-14C	16,000	テキサスインスツルメント	DS990-Model-S	13,800

※Ergonomics（人間尊重の立場から、人間の安全と衛生を確保し、人間と機械との調和をめざす科学。）

ERGONOMICSを追求する

MAY FLOWER

株式会社 明光製作所
〒173 東京都板橋区熊野町35-11
TEL.(03)955-2231(代表)

フロッピー・ディスクを無駄にしていないか!

自作派のためにプロのノウハウを公開 **"ファイル入門"**

本格業務のために即戦力となるソフト **"Condor S-20"**

"リレーショナル・データ・ベース"

フロッピーディスクはデータの宝庫です。フロッピーディスクを手にしたとき、あなたはパソコンの未知の扉をまた1つ開くことになるのです。そこはデータ・ベースの世界です。

多くの人が、その扉に挑戦してきました。そして大多数の人が、その扉の固さに降参してきました。そこはプロの世界なのです。そしてあなたの武器はマニュアルにある数個のコマンドだけなのです。あなたのデータ・ベースの城を構築するまでには、幾多の難関を突破しなければなりません。

データ・ベース……。抽象的で難かしい概念ですが、自分が何をしたいのかを明確にすれば、きわめて、具体的な話になります。自分が日常扱っているデータを入力すること、入力したデータをどう計算するか、そして、自分のイメージに合わせてプリントするかに尽きます。

「ファイル入門」では成績管理を題材にしています。自分が学校の先生になったと思って下さい。生徒の台帳を持っていて、夜な夜な試験の成績表を作成したり、順位ずけをしたりに追われて、こんなことはコンピュータでもできるのに…、自分は生徒に与える問題を考えたり、もっと生徒のことを考える時間が欲しいのに……と考えている。パソコンを買い、フロッピーディスクを買って、何とかしようとい状態を想定していただきます。

「ファイル入門」では、成績管理を題材にしてデータファイルに対して、データの入力、修正、プリント、ソート、検索についてのプログラム例をコメントつけて説明しています。クラス毎の生徒の入力、試験の点の入力、点数順のソート、合計点の計算、点の悪い人の検索、リストの作成を行います。

「ファイル入門」を作るにあたって、汎用のデータ・ベースを作るのは難しいけれど、自分専用のデータ処理を行うプログラムを作るのは、比較的簡単だと思いました。その理由は、一度パターンを作ってしまえば、後はその項目を変えるだけで応用ができるからです。

パソコンでファイル処理を行うには、高等テクニックも必要です。インデックス・ファイルの作り方、グループ毎のインデックスの作り方をマスターしますと限りあるフロッピーディスクをより有効に使い、処理時間を短縮することができます。「ファイル入門」はワンタッチで、自分のプログラムをジェネレートするプログラムではありません、「ファイル入門」に含まれているプログラムをLoadして、自分にあった項目名、フィールドに変更して、ファイル処理を理解しながらプログラムを作っていく学習プログラムです。私達が作ってきた実務ソフトの経験を生かした、実践のための学習プログラムです。あなたの未知の扉を開くガイド・ブックです。

「ファイル入門」で学習し、自分でファイル処理のプログラムを作る余裕のない場合はどうしたらよいかという質問をよく受けます。本格的なデータ・ベースはそのために用意されています。精密なデータの城を作るには、自作のファイル処理プログラムではやりきれません、もし、あなたが業務用のプログラムを作ろうとして、ファイルの世界に入り込んでいるとしたら、Condor S-20をおすすめします。Condor S-20はリレーショナル・データ・ベースで、はじめてファイルを扱う人にも使いこなせるデータ・ベースです。リレーショナルであるべき理由は、はじめに作ったデータ・ファイルを後から自由に改造できることです。使う人の発想に自由に付いてきてくれるソフトがリレーショナル・データ・ベースであり、その中でも使い易さ抜群のデータ・ベースがCondor S―20です。

簡単な処理なら自分でプログラムを作るという人のために、「ファイル入門」パッケージによって、ファイル作成のための道しるべを提供しています。業務用のプログラムで、かなりの量のデータを扱わなくてはならない人のために、Condor S-20をおすすめします。いずれにしても、自分のイメージするファイルの形をより具体的に実現できるか、より手軽に作れるかということが、私とあなたの共通のテーマになると思います。

価格
ファイル入門 ¥18,000 FM-7.8.11
Condor S-20 ¥260,000 FM-7.8.11.他

C.C.W PACKAGES
FM-7,8,11

BASIC ジェネレータ
¥18,000円(5D)
文番号の自動置き替え、サブルーチン間の受け渡し、変数の自動置き替えにより、複数のプログラムの連結を行います。

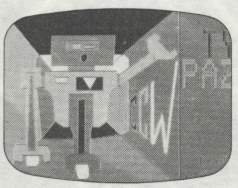
プログラマーズ"パル"
¥18,000(5D)
BASIC言語で作成されたプログラムを、コロン(:)、FOR NEXT、IF THEN(ELSE)の段落にきって表示します。さらに指定によってプログラム中の使用変数、使用サブルーチンとび先の一覧を作成します。

三次元座標基本ルーチン ¥45,000
2ベクトルのなす角、2平面の交線と交角、座標交換と点の回転、多点入力による球、球と平面の交線等17種類のルーチンから成っている。

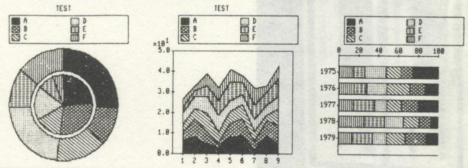
等高線作画 ¥100,000
YY一平面に分布する任意量測定又は計算結果の等高線(等値線)を描く。

セブンがあれば卒業できる
電訳パソコン英語編
FM-7用
¥9,800 (5D)
近日発売
英和・和英瞬間検索可能
英単語約4,700語収録

おもしろタイプ英文編
FM-7用
価格¥3,500(テープ)
はじめてキーボードに触れる方も楽しみながら覚えてしまうおもしろ学習プログラム



グラフパッケージ FM-7.8.11価格¥35,000(5D)
●グラフ種類 棒グラフ, 積上グラフ, 連立グラフ, 折線グラフ, 棒・折線グラフ, 積上・折線グラフ, 連立棒・折線グラフ, 相対値帯グラフ, 積立折線グラフ, くもの巣グラフ, 円グラフ, 二重円グラフ



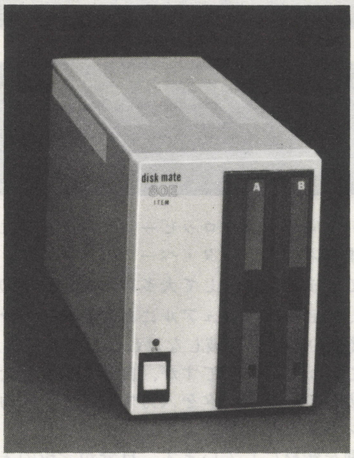
ご注文は現金書留又は銀行振込でどうぞ(銀行振込の場合は、はがきであらかじめご注文下さい)○振込先 **大和銀行新宿支店普通6108825**

CCW 株式会社 **シー・シー・ダブル**
〒151 東京都渋谷区代々木 2-21-10山佐ビル4F
(03)379-1831

●CONDOR講習会開催● Condor S-20 ガイドブック¥700(送料¥170)
当セミナーはCONDOR S-20をより効率的にご利用いただくためのものです。
会場:〒151 東京都渋谷区代々木2-21-10山佐ビル4F
株式会社シー・シー・ダブル内 CONDORセミナー係
連絡先:03-379-1832

どこまで広がるか
コンピュータライフ。
SOFT新時代にむけて、
いまITEMから……

NEW disk mate 80Ⓔ 5インチデュアル フロッピーディスク



■disk-80シリーズの普及機
■大きさも一周り小さくなって よりコンパクトに
■リニアシーク機構採用
■このクラスで初めて10万円を 切って新登場！

disk mate 80Ⓔシリーズ

■80Ⓔ-1 (NEC PC用) …… 108,000円
■80Ⓔ-2 (FM-7用) ………… 98,000円
■80Ⓔ-3 (X1用) ………… 108,000円
■PC用専用ケーブル……… 5,000円
■X1用, FM-7用 ………… 12,000円 (インターフェイス・ケーブルセット)

98,000円 (標準仕様)

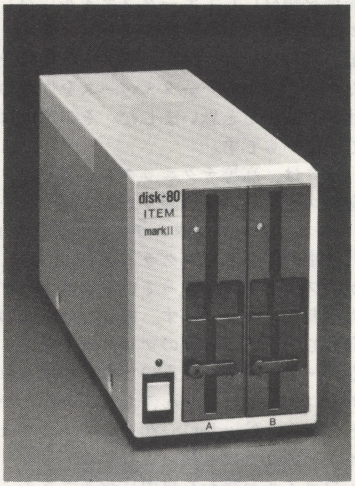
今日からは Disk! 5インチシングル フロッピーディスク

58,000円 (M10-9標準仕様)

disk mate M10シリーズ

■M10-1 (NEC PC用) ……… 69,800円
■M10-2 (FM-7用) ……… 69,800円 (インターフェイス込)
■M10-3 (X1用) ………… 78,000円 (インターフェイス込)
■M10-9 (2台目用) ………… 58,000円
■専用ケーブル ………… 5,000円

快走！ disk-80markⅡ 5インチ高速デュアル フロッピーディスク



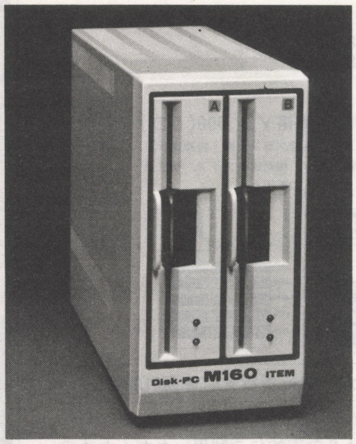
■高速！そのうえ静寂
■メカ部に定評の松下通工製 JA551を採用
■トラック間アクセスタイム 6m secを実現

■disk-80PⅡ (NEC PC用) 123,000円
■disk-80FⅡ (FM-8, 7用) …… 128,000円
■disk-80BⅡ (MZ用) …… 138,000円
■専用ケーブル ………… 5,000円
FM-8, 7用 I/O ………… 15,000円

123,000円 (disk-80PⅡ本体)

実績No.1 パワフルディスク 8インチデュアル フロッピーディスク

Disk-PC M160

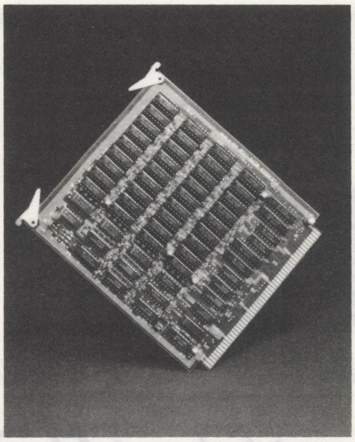


■Disk-PC M160-01 … 256,000円 (PC-9801用)
■Disk-PC M160-02 … 298,000円 (PC-9801用 CP/M86付)
■Disk-PC M160-03 … 296,000円 (PC-8801, PC-8001mkⅡ用)
■Disk-PC M160-04 … 296,000円 (PC-9801E, F用。インターフェイス含)

256,000円 (専用ケーブル付)

NEW RAMディスクも夢でない！

PC-9801E, F用 増設メモリーボード M256



■256KB実装済
■PC-9800シリーズに そのままダイレクトイン
■最大512KBまでの増設が できます。
■パリティエラーチェック 機能付

69,800円 (256Kバイト実装済)

株式会社アイテム
〒251 藤沢市南藤沢8-1-202
TEL 0466-27-1668(代)
FAX 0466-27-2800

■お買い求めは全国マイコンショップで。
■入手できない方は通販を行なっております。 直接当社にご連絡下さい。

*CP/M®はデジタル・リサーチ社の登録商標です。*UCSD Pascal™はカリフォルニア大学理事会の登録商標です。*FLEX®はTSC社の登録商標です。

GRAPHIC CANVAS
FMユーザー待望の
グラフィック・パッケージ
新・発・売
グラフィック・キャンバス
夢、発進
FM-7 FM-8 FM-11共通
5インチ ￥9,800
カセット ￥3,900
デジタイザー対応!!
(K-510)
■簡単に中間色(3色までの組み合わせ)を塗ることができます。
■中間色は全60色(40色は登録済。20色はユーザ登録可能です。)
■操作はメニュー方式によるコマンドの選択で非常に簡単に使用できます。
■グラフィックの入力方法はデジタイザーとキー入力(カーソル移動)が選べます。
■作成したグラフィックはマシン語で実行！高速処理が可能です。
企画・販売元 株式会社 アイ企画
〒530 大阪市北区天神橋2-4-17 千代田第1ビル8F ☎06-352-7137
東京営業所:〒170 東京都豊島区南大塚3-34-7 丸善大塚ビル ☎03-981-3701
開発元
CSKソフトウェアプロダクツ/HOT B
[広]155
※広告の内容は変更することがあります。

FM-7/8

FMS

FILE-MANAGEMENT-SYSTEM

今までのデータベースとココが違います!!

今までのカード型データベースに、あきたらないあなたへ
情報処理にムダな時間をついやしていませんか?
FMSは、そんなあなたに、ほんのわずかな時間でお役に立てます。今までのデータベースでは、各種の制約に悩まされましたが、FMSでは、そんな悩みは不要です。メモ帳感覚で使っていただけます。それにディスク1枚に2,048件ものデータ管理ができ、今まで出来なかった大量データ処理も可能になりました。

〔特長〕
- ●画面は完全フリーフォーマット・書式設定が、自由自在。
- ●1レコード最大32ページ・文字数では約56,000字。
- ●データ件数最大2,048件・大量データ処理可能。
- ●強力検索機能・多重検索、文字列の一部でもOK。
- ●高速ソート機能・英字、五十音、数値OK。
- ●項目設定　自由自在。(設定制約なし)
- ●レポート機能・計算機能を完全サポート。
- ●プリンターへの印字は書式自由。

御注意

・ノート ・レポート ・計算機能につきましては
オプション(近日発売予定)にて提供させて頂きます。

マニュアル・特製バインダー付き
5インチ ¥29,800
マニュアル別売り致します。

Policy 株式会社ポリシー

〒542 大阪市南区南炭屋町43 アベニュー心斎橋 TEL 06(211)8987

■皆様からの御意見・御要望をお待ちしております。
■お求めは、有名パソコンショップか、現金書留で住所、氏名、商品名、機種名明記の上、左記までお送り下さい。
尚、送料として350円加算して下さい。

ますます

コンピュータ・イレブンオリジナルソフト&書籍

充実!!FM用ソハコムブランド。

全国の
パソコンショップ
でお求め
ください。

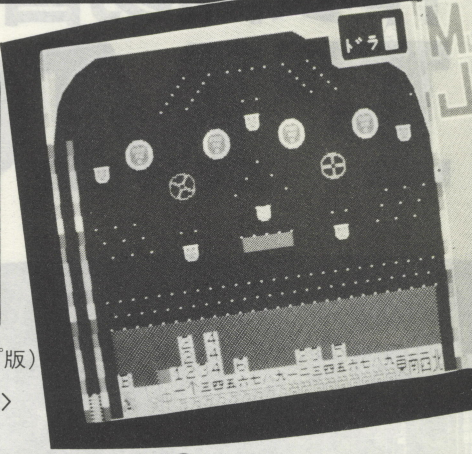
新発売

muscat 雀球 Jan-Q

〈世界で初めて!?〉

オールマシン語だから超高速!玉の動きも部品の動きも雀球そのまま! あなたはパチンコ屋で味わう以上の面白さ、楽しさを感じるだろう!FM-7の壁を打ち破る画期的ソフト遂に完成!!

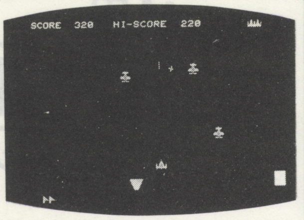
¥3,800(テープ版)
〈ディス版も発売予定!〉
(FM-7用)



EXPLORER

¥3,300(FM-7用・テープ版)

宇宙探索に向う"EXPLORER"号の目前に広がる大宇宙と未知の惑星。突如、来襲する謎のUFO船団の攻撃に、"EXPRORER"号は立ち向って行く。さあ、あなたは幾つの難局を乗り越えられるか!?広大な宇宙探索の目標を達成できののはいつの日であろうか?
※乞う、ご期待!!PART2、PART3、(近日発売)

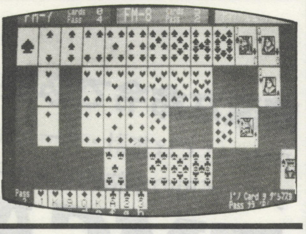


欲ばりな君に! **¥3,300**

2倍楽しめる トランプゲーム

FM-7用(テープ版)

ポピュラーな七並べ、スリリングな知的ゲーム、ツーテンジャックができます。



ルンルン作曲

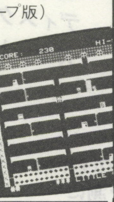
¥3,300(FM-7/PASOPIA-7用)(テープ版)

★キーボードが鍵盤に!もちろん半音階も★2オクターブ半の音域でらくらく作曲★自動演奏で編曲もワンタッチでOK!!

リトルギャング

¥3,300(FM-7用・テープ版)

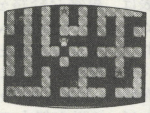
ホワイトキャットはすご腕のリトルギャング!家の中の金目の物を全て盗み出すと次の家へ。トランポリンとドアをうまく利用して、だんだん複雑になる家の中をカッポし、最高点を盗みとろう!



ペンギンくん

¥3,300
(FM-7用・テープ版)

「ペンギンくん」の行く手に現われる怪物を、南極の氷片を蹴とばして退治する。次々に変る場面ごとに怪物を退治しないと危うし!!「ペンギンくん」、サアあなたは何点得点できるか!?



SOHACOM(ソハコム) 今、話題の

ケイ線描きよサヨウナラ!めんどうな計算もゼロ!

株価チャート分析システム

¥100,000 TYPE II

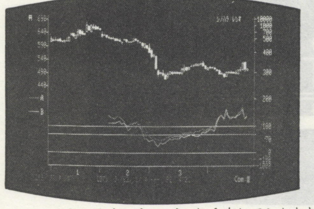
三菱MULTI-16・PC9801 8801・FM-7 8用
●基本システム
¥100,000〜¥150,000
●フルシステム ¥300,000

FM-7・8・11
PC8801・9801
PC9801F
MULTI-16
IBM5550用

基本システム(8種) ローソク足・移動平均・出来高・信用取引残高・カイ離度・相対レシオ・新値三本足・カギ足

追加システム(オプション、5種) 逆ウォッチ曲線・ボリュームレシオ・OBV線・篠原レシオ・一目均衡表

★新発売システム★
Ⓐサイコロジカルライン…¥30,000
Ⓑポイント&フィギュア…¥30,000
Ⓐ+Ⓑ=~~¥60,000~~→¥50,000



※オンラインサービスも予定しております。

SOHACOM(ソハコム)

グラフ化ノンプログラミング

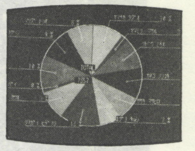
MDB11

¥35,000〜¥50,000
漢字対応型〈FM-8用〉は1万円高となります。

■対応機種
MDB11
if800
PC-8001
PC-8801
PC-9801
FM-7
FUJITSU MICRO 8
レベル3
MULTI16
FP1100
MZ3500用

——新登場——
PC-9801用
(PC-9801F対応)

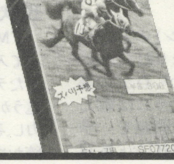
お蔭様で
販売実績
5,000本突破!
ノンプログラミング汎用
データベース
簡易言語



❶ソースリスト全公開。340ページのマニュアルに内部徹底解説。
❷1,000件のデータの中から1件の検索5秒。(高速インデックスサーチ機能)
❸作表、グラフ化機能付、強力な報告書(レポート)作成機能。
❹データファイル処理のプログラミングは不要。
❺分類・並べ替えやトランザクション処理も自在。

ズバリ予想!! 実戦競馬

(テープ版) **¥3,300**
(FM-7
PC-8001mk II
PASOPIA-7用)



英文・カナ タイプ練習

(テープ版) **¥3,000**
(FM-7 8・PC-8801
PC-8001mk II
PASOPIA-7用)

数値計算 パッケージ

(テープ版) **¥4,800**
(PC-8801・PC9801
FM-7用)

求 ★あなたの開発したソフト求めます!!
★あなたのアイデアを本にしませんか!?

あなた独自のソフトを商品化したい、買い取ってほしいとお考えの方、又はあなたが書かれたパソコンのノウハウを本にしたい方、下記までご連絡下さい。あなたもベストセラーをつくりだしてみませんか?

お問合せ先/(株)日本ソフト&ハード社出版部まで。
〒160 東京都新宿区高田馬場2-8-3・生沼ビル2F
☎03(209)2585(代)

四、八〇〇円の日本語ワープロソフト。個人ユースにモテモテ。

「ザ・FM漢字」

(FM-7・8用)
カナ漢字変換入力。
ヨコ書きもタテ書きもOK。
テープ版 **¥4,800**
ディスク版 **¥6,800**
エプソンプリンタ(MP-80タイプIII、FP-80、RP-80)でプリント可能。

SOHACOM(ソハコム)
ソハコムブランド BOOK シリーズ

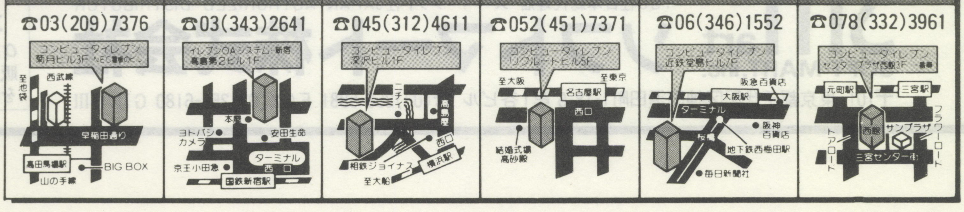
★マシン語ゲームのつくり方(PC8001、FM-7·8、L-III用)…………¥2,800
★FM-7·8用マシン語の本…………¥2,800
★6809アセンブラ…………¥3,800
★6502アセンブラ…………¥4,000
★パーソナルコンピュータのデータ処理…………¥2,800
★「おもしろまじめ」F-BASIC…………¥2,800
★最新刊 X-I·X-IC·X-IDマシン語入門…………¥2,800

刊行予定
「FM-7プログラムライブラリー」
3月上旬発売予定

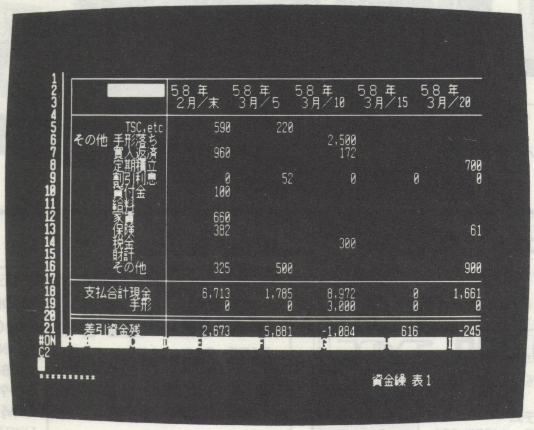
コンピュータ&データコミュニケーション
コンピュータ・イレブン

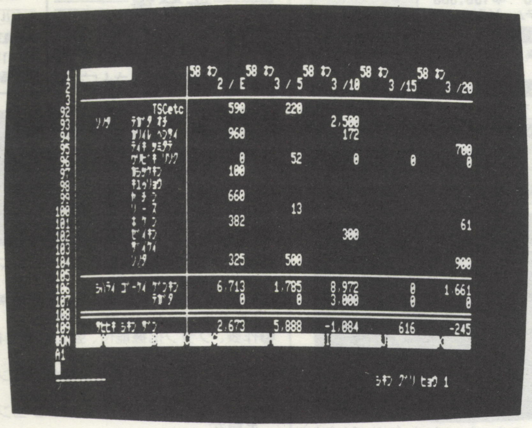
●高田馬場本店/東京都新宿区高田馬場2-17-4 菊月ビル3F ☎03(209)7376(代)
●新宿支店/東京都新宿区西新宿1-9-13 高倉第2ビル1F ☎03(342)4821(代)
●横浜支店/横浜市西区南幸2-5-4 深沢ビル1F ☎045(312)4611(代)
●名古屋支店/名古屋市中村区椿町1-16リクルート名古屋ビル5F ☎052(451)7371(代)
●大阪支店/大阪市北区堂島2-2-2 近鉄堂島ビル7F ☎06(346)1552(代)
●神戸支店/神戸市中央区三宮町2-1-5センタープラザ西館3F ☎078(332)3961(代)

高田馬場店	新宿西口店	横浜西口店	名古屋駅前店	大阪梅田店	神戸三宮店
☎03(209)7376	☎03(343)2641	☎045(312)4611	☎052(451)7371	☎06(346)1552	☎078(332)3961



作表簡易言語シリーズ
FMCALC





待ってました!!

漢字FMCALCはFM-11添付ソフトの「FMCALC」のパワーアップ版です。もちろんデータも共有できます。

新登場

FM-11用
漢字FMCALC
ディスクバージョン……¥58,000

簡潔な命令で必要十分な処理を!!

FMCALCは集計・ソート作業に威力を発揮するFMシリーズ用の本格的ビジネスソフトです。

FMCALC FM-7用 FM-8用
ディスクバージョン……………………¥48,000
カセットバージョン……………………¥15,000

FMCALCは管理職の方のためのパーソナルツールです

今日、OA(オフィスオートメーション)の普及により大量のデータ処理が可能となりました。しかし一般管理職や中堅技術者の方々が、自分の業務のためにコンピュータを個人的に使うには難しい問題があります。FMCALCはこの問題を一挙に解決。FORTRAN、COBOL、BASICなど電算用の言語を学ばずに、すぐさまデータを入力し、機械に計算させ、結果を画面から読みとり、またプリントアウトすることができます。

FMCALCは作表専用ツールです

管理業務には沢山の数値データを取り扱うことが避けられません。また多くの場合、数値データを表の形にまとめ、縦横の方向に和をとったり、平均したり、構成比を算出したりすることが必要となります。FMCALCは2〜3のキー操作をするだけで、白紙の表を画面に表示します。あなたは画面を見ながら必要な数値や文字や式を必要な場所に次々とキーボードから入れていくだけで、希望の表を作ることができます。

現状の把握や将来の予測に

FMCALCで作成された表は、複雑な経路を辿って入力されたデータではなく、あなたが直接入力したデータに基づいています。たとえば実数値とは違うが、将来の予測に役立ちそうな数値を試みに入力し、その結果、その他の数値がどんな風に動くかを確かめることができます。したがって販売努力の目標設定や、資金繰の予測などに威力を発揮します。

FMCALCは多様な可能性を秘めています

FMCALCは簡便なソフトウェアですが、深みのあるプログラムです。あなたの創意と工夫を十分に受けとめ、縦横にその能力を発揮する可能性が秘められています。プログラミング言語という媒介を経ずに、あなたは直接いろいろな数値をプランニングし、FMCALCに指示することができるのです。

FMCALCワークシート新発売!

●FMCALCをさらに使いやすくするように作られました。
●FMCALCとワークシートによって、貴方のFMシリーズを本格的ビジネス用パソコンとして、その能力を縦横に発揮できます。

FMCALCワークシート1(売掛管理表、買掛管理表)……¥9,800
FMCALCワークシート2(手形管理表)……¥9,800
FMCALCワークシート3(資金繰表)……¥9,800
★ワークシート1,2,3セット価格……¥25,000
★CALC-PACK ワークシート1,2,3+FMCALC……¥68,000

SMart SOFT MART, inc.
TSC社日本総代理店・スーパーソフト社 JAPAN AUTHORIZED DISTRIBUTOR
ソフトマート株式会社
〒101 東京都千代田区神田須田町1-18-6 第1谷ビル ☎(03)256-5881 FAX(03)256-6180 GII/GIII

通信販売のお知らせ

α言語、FMCALCをはじめ取扱商品の通信販売をいたします。品名、住所、電話、氏名を明記の上、現金書留にてお申込み下さい。

PCA SOFT
手軽に活用パソコンOA

僕たちだって働きたい

やる気十分のあなたのパソコン。そんなパソコンを遊ばせてはいませんか。PCAのソフトはユーザーの助言を生かして使いやすさと信頼性と、スピードを追求したシステムです。PCAで働くパソコンに変身させてあげてください。

まさに即戦力！ビジネスソフトの決定版です。

商魂（販売・売掛システム）

●**手軽な操作と安全設計**。誰にでも使えて高機能。●得意先、商品件数は**合計1500件**（5インチ2 DD1000件）。両者の組み合わせはユーザーが任意に設定。●登録外の得意先、商品名でも自由に入力でき、納品書発行も可能です。●適応範囲は1取引10億円未満、月商100億円の企業まで、幅広い業種に対応。●売価や値引きは、金額、値引率の両方で設定。●出力帳簿はヒサゴのコンピュータ用紙を使用。●納品先が複数でも請求書統括処理が可能です。●会計帳表や豊富な販売管理資料（14表）が必要な時に作成できます。●請求書の発行や締切処理に関係なく、締日の翌日から伝票発行が行えます。●宛名ラベル印刷などの実務的な機能が豊富。●その他、合計で20の特長、オフコン並の機能。

使用機種　ＰＣ－9801Ｆ/Ｅ
システム価格　￥98,000

ザ・パソコン会計

●家計簿程度の知識で**誰でも使える**操作性、会計処理が格段にスピードアップ。●いま使用中の伝票や出納帳をそのまま活用。●主要48科目は**ワンタッチ入力**。●入出金データは**自動仕訳**で正確に管理。●データの訂正、**追加**、**削除**も簡単にできます。●**日付単位**の期間指定、科目指定が可能。●1500社以上の**豊富な販売実績**

●**豊富な対応機種**：漢字版　ＰＣ－9800、　ＦＭ－11、ＳＰ－200、ＪＢ－3000
カナ版　ＦＭ－7、8、11

〔機能〕 主勘定科目（240）漢字7文字、補助科目（240）漢字7文字、摘要文漢字14文字、摘要文コード登録（200）漢字7文字

〔ソフト価格〕
・システムA（元帳、試算表、決算表）……￥ 98,000
・システムB（A＋要約日計表、補助簿、手形管理表）￥148,000
・システムC（B＋部門管理、各種分析表）……￥198,000

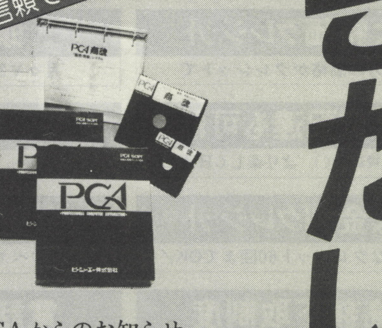
ザ・パソコン給与

●コンピュータの知識は不要、**使いやすさ**は抜群。●就労基準、支給項目、控除項目等はユーザーに合せて使えます。●給与計算から年末調整まで**連結処理**も可能。●毎月の給与計算は、残業時間などの**変動項目を入力するだけ**でＯＫ。●データの訂正、**追加削除**も容易に。●個人別、グループ別のデータ集計も可能●住民税の納付先一覧も作成。●ザ・パソコン給与IIは銀行振り込み、端数処理など**各種人事管理資料**も作成。

〔機能〕 給与I・対象人数（200）・対象者（月給、日給、時給）
給与II・対象人数（500）・対象者（月給、日給月給、日給、時給）

〔ソフト価格〕	給与I（ＰＣ－9800）	給与II（ＦＭ－11）
・給与計算編（6表）	￥59,000	￥148,000
・賞与計算編（5表）	￥39,000	
・年末調整編（5表）	￥39,000	￥ 39,000

やさしさと信頼でうけてます!!



——PCAからのお知らせ——

●実践パソコン教室

『実践パソコン会計』　3月21日(水)13：30～17：00
『実践パソコン給与』　3月27日(火)13：30～17：00
場所・問い合わせ先　オムロン・マイコンベース銀座　tel 03-535-3381

ピーシーエー株式会社

〒160 東京都新宿区西新宿7-7-33(新銘ビル新館2F) TEL03(368)9631(代)
〒530 大阪市北区中崎西2丁目2-1(東梅田八千代ビル) TEL06(315)8637(代)
〒451 名古屋市西区菊井2丁目6－7 TEL052(565)0406(代)

販売代理店募集中 電話かハガキで三島か岡田まで

ご来社テスト自由、資料請求は切手200円分を同封の上資料請求券を添えてお申し込み下さい。

資料請求券
FM
4月号

ぐ〜んと春、じ〜んとくるお値段ツクモ全店

合格おめでとう 激安 セール！

4/10まで

頭金ゼロ・毎月のお支払いもゼロ
お支払いはボーナス2回払い！
ツクモウルトラクレジット

ツクモトレードセンター大好評！

貴方がパソコンを売りたい・買いたいと思った時、ツクモトレードセンターにお問い合わせ下さい。当センターで動作状態が完動か整備かをチェックし、商品を充分査定の上お客様の希望価格を設定してツクモトレードセンター中古品リストに登録致します。

お電話かニューセンター店にてお申し込み下さい

1 受付

売りたい人 直接品物をお持ち頂いても結構です。その場で打ち合わせをし、トレードセンターお客様カードに内容等記入致します。お電話でお申し込みになった場合もお客様カードを記入し、送付致しますので条件確認の上、サイン及び捺印して頂き申し込み金￥2,000（トレード成立後返金致します）を同封の上、返送して下さるようお願い致します。

買いたい人 委託者同様ニューセンター店又は電話で受付致します。お客様カードを承認の上、記入・捺印して頂き、送付してもらいます。

2 取引方法 確認されたお客様カードが着き次第御連絡致します。その時に商品を当センターに送付して下さい。ツクモトレードセンター技術スタッフが動作状態等査定し、再度御連絡して希望価格を設定致します。（この場合、販売価格の変更も有りますので御了承下さい。）

お預かりした商品は店頭又は電話にて購入希望者を受け付け致します。希望先着順にお電話で商品内容を出来るだけ細かく説明致します。

■査定基準

1 メーカー及び機種名、S/No.セットバージョン ②使用年月 ③動作状態（完動か整備が必要か） ④マニュアル、ケーブル等付属品の有無 ⑤オリジナルダンボール及びパッキンの有無

■トレード成立後代金は……

委託者の条件で手数料（含む振り込み手数料）を差し引いた金額をすみやかに振り込みます。

■梱包状態不良による輸送中の事故は当社では責任を負いかねます。

ツクモトレードセンター☎03-251-9977

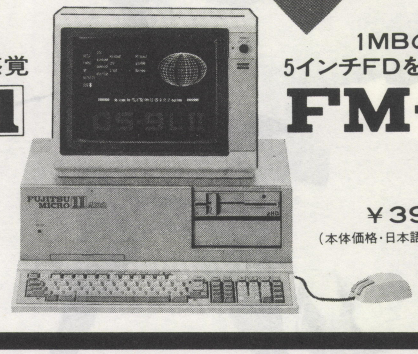
だからツクモで買う…
ツクモウルトラシステム

大特価クレジット 魅力の価格がクレジットで	**頭金ナシ** 月々￥3,000クレジット
日曜配達も可能 （札幌店除く） より楽しく日曜日	**支払いは約2ヶ月後** スキップクレジットで
低金利クレジット 便利なクレジット60回までOK！	**全商品保証付** 製品はすべてメーカー保証付
高額下取制度 ツクモ全店高額下取りOK！	**電話クレジット** お申し込みは電話一本でOK！

★今、お申し込みになればお支払いはあなたのご希望に（但し6ヶ月以内）

★ツクモらくらくクレジットお問い合わせは☎251-9911まで

富士通新製品



OS-9を採用、ビジネスもマルチ感覚

FM-11 AD2

￥298,000

（本体価格・日本語ワープロソフト付）

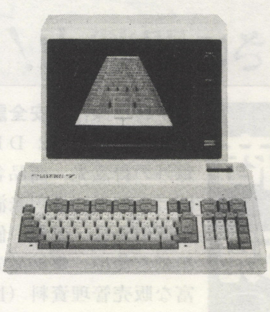
1MBの大容量5インチFDを2基搭載

FM-11 BS

￥398,000

（本体価格・日本語ワープロソフト付）

富士通FM-7 組み合わせ 本体標準価格￥126,000



★FM-7組み合わせ

Aセット FM-7（本体￥126,000）＋TVアダプター（￥12,600）＋データレコーダ（￥12,800）＋オリジナルソフト5本（￥15,000）
＝合計定価￥166,400➡特価**￥125,800**

Bセット FM-7（本体￥126,000）＋TH14NM2G（14″1000文字 カラーCRT￥69,800）＋ケーブル（￥1,500）＋データレコーダー（￥12,800）＋オリジナルソフト5本（￥15,000）
＝合計定価￥225,100➡特価**￥148,800**

Cセット FM-7（本体￥126,000）＋NH14DD（14″2000文字カラーCRT￥69,800）＋ケーブル（￥1,500）＋データレコーダー（￥12,800）＋オリジナルソフト5本（￥15,000）
＝合計定価￥225,100➡特価**￥167,800**

Dセット FM-7（本体￥126,000）＋NH-12D9（12″2000文字カラーCRT￥99,800）＋ケーブル（￥1,500）＋データレコーダー（￥12,800）＋オリジナルソフト5本（￥15,000）
＝合計定価￥253,600➡特価**￥175,000**

Eセット FM-7（本体￥126,000）＋C14VC1（2000文字ファインピッチRGB付TVモニター￥93,000）＋ケーブル（￥1,500）＋データレコーダー（￥12,800）＋オリジナルソフト5本（￥15,000）
＝合計定価￥248,300➡特価**￥179,000**

★FM-7＋プリンタ組み合わせ

Gセット FM-7（本体￥126,000）＋NH12D9（12″2000文字 ￥99,800）＋KP-3000（プリンター￥84,000）＋プリンターケーブル（￥6,000）＋オリジナルソフト5本（￥15,000）＋データレコーダ（￥12,800）
＝合計定価￥345,100➡特価**￥249,000**〈プリンター用紙サービス中〉

★FM-7＋DISK組み合わせ

Hセット FM-7（本体￥126,000）＋NH12D9（12″2000文字￥99,800）＋KD290/FM（￥148,000）・K-305FM（￥128,000）・NF-210FM（￥138,000）・TF10F（￥125,800）のうちいずれか1点＋ディスクインターフェースケーブル（￥15,000）＋システムディスク（Disk BASIC￥7,000）＋オリジナルDISKソフト6本入り1枚（￥9,800）＋TS2D10枚（生ディスケット￥15,000）
＝合計定価￥395,400〜￥423,600➡特価**￥310,000**

★FM-7＋ワープロ組み合わせ

Iセット 【TAPE版ワープロ】FM-7（本体￥126,000）＋NH14DD（14″2000文字カラーCRT￥69,800）＋カラーCRTケーブル（￥1,500）＋データレコーダー（￥12,800）＋KP-3000（プリンター￥84,000）＋プリンターケーブル（￥6,000）＋F漢（TAPEワープロソフト￥7,500）＋オリジナルソフト5本（￥15,000）＋漢字ROM（￥35,000）
＝合計定価￥357,600➡特価**￥274,800**〈プリンター用紙サービス中〉

各定価￥2,500をプレゼント!!

★FM-7お買上げの方
RAM DISK プログラム（裏RAMをディスクとして使うソフト）

★FM-7プリンタお買上げの方
プリンタバッファプログラム（裏RAMをプリンタバッファとして使うソフト）

★FM用ディスク又はFM-11お買い上げの方
インターリーブフォーマットコンバータ（ディスクのアクセスが2〜3倍速くなるフォーマットプログラム）

通信販売の受付は
03(251)9911へ
ご注文の際は、ご注文になりたい商品名をご記入の上、現金書留にてお願い致します。20万円以上は銀行振込みも取り扱っております。お問い合わせ下さい。
富士銀行 神田支店
普通口座 No.894047

即決 ツクモ全国クレジットOK!!
- 現金特別価格でツクモクレジットが利用できます。残金のみに金利がかかります。
- 60回払いまで出来ます。但し、1回の支払い額は3,000円以上。
- その場で、お持ち帰りできるクレジットもあります。
- 印鑑、身分証明書（免許証など）が必要です。

●取扱いクレジットカード：日本信販、JCB、DC、UC、VISA、MC、CF（尚、クレジットカードご使用の場合は店頭員にご相談ください。）

■ご注文商品の中、未着の品物がございました時は、通販受付電話で納品書No.をお知らせ下さい。尚、商品の詳しい取り扱い等は、各販売店までお問い合わせ下さい。

■通信販売は 〒101 東京都千代田区神田郵便局私書箱135 九十九電機㈱Oh！FM4係

九十九電機株式会社

ニューセンター店	〒101 東京都千代田区外神田1-16-10	☎03(251)0986〜7
名古屋店 3F	〒460 名古屋市中区大須3-30-86 ラジオセンターアメ横ビル内	☎052(263)1681
5号店	〒101 東京都千代田区外神田3-1-14	☎03(251)0531〜2
7号店	〒101 東京都千代田区外神田1-9-9	☎03(253)4199
札幌店	〒060 札幌市中央区南二条西3-15-1 さっしんビルB1	☎011(241)2299
ビデオセンター店	〒060 札幌市中央区南二条西4 乙井ビルB1・1F	☎011(241)3999

■定休日 東京各店は毎週木曜日と第3水曜日、名古屋店は毎週月曜日
札幌店（マイコン・ビデオ店）は第3木曜日

★セットの場合は梱包送料￥2,000を加算して下さい。

あなたもプログラムデザイナー！

●第１回●
ボーステック　プログラムコンテスト

賞金総額　**300万円**

★　１等　　100万円……………………〈１名〉
★　２等　　50万円……………………〈２名〉
★　３等　　10万円……………………〈10名〉

募　集　要　項

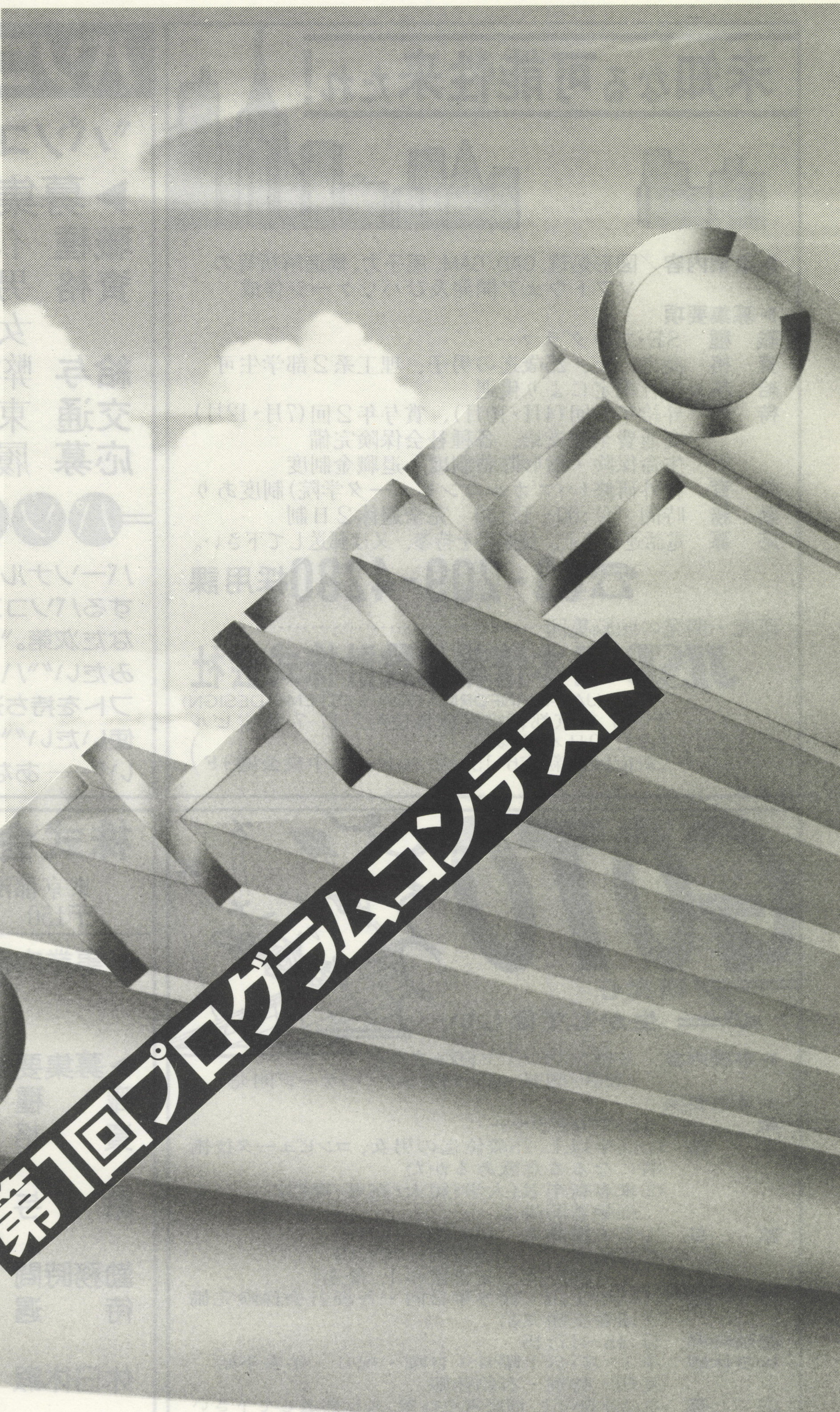
●募集期限
昭和59年５月10日（当日消印有効）

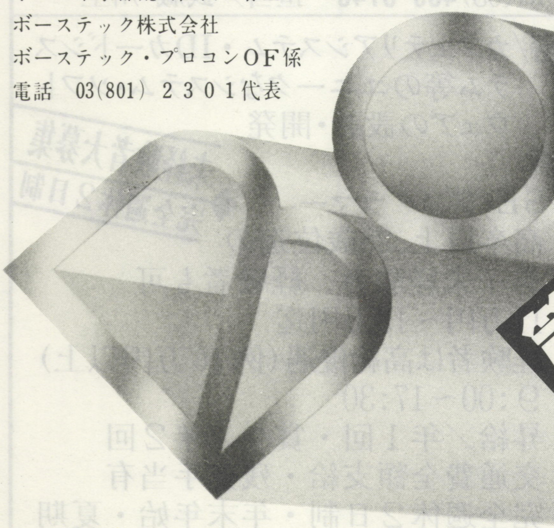
●募集内容
未発表のオリジナル作品であれば、ゲーム、ビジネス、ユーティリティーなど種類は問いません。使用機種は、市販されているパソコンであれば特に限定しません。

●応募方法
個人・グループで、何点でも応募できます。プログラム名と氏名を明記したテープ又はディスケットとともに、〈１〉氏名(フリガナ)〈２〉性別、年齢〈３〉住所、勤務先・学校名、電話番号〈４〉プログラム名〈５〉使用機種・言語〈６〉プログラムの特徴・操作方法 を添えて（Ａ４サイズ）お送りください。
応募作品は返却いたしませんので、あらかじめご了承ください。入賞作品は商品化し、賞金の他当社規定の印税をお払いします。

●発　　表
発表日……昭和59年５月31日
毎日新聞、各社マイコン誌。入賞者には直接通知します。

●応募・問い合わせ
〒116 東京都荒川区西日暮里２－41－16
ボーステック株式会社
ボーステック・プロコンＯＦ係
電話　03(801)　２３０１代表





BOTHTEC inc.　ボーステック株式会社
〒116 東京都荒川区西日暮里2-41-16 TEL (03)801－2301(代)

S・Eプログラマーをめざす君へ

未知なる可能性来たれ!

▶事業内容／図形処理、CAD/CAM、原子力、構造解析等の
ソフトウェア開発及びパッケージ作成

▶募集要項

職　種　SE・プログラマー
資　格　高卒以上、25歳迄の男子、理工系２部学生可
給　与　当社規定により優遇
待　遇　昇給年２回(4月・10月)、賞与年２回(7月・12月)
交通費全額支給、各種社会保険完備
生命保険・財形貯蓄制度・退職金制度
教　育　社外研修(パスカルコンピュータ学院)制度あり
勤　務　時間／９:00～17:30　完全週休２日制
応　募　電話連絡の上、履歴書を持参、又は郵送して下さい。

☎03-209-4280採用課

研究・開発の頭脳集団……………………………

JISD 日本情報化設計株式会社
(JAPAN INFORMATION SYSTEMS DESIGN)
〒160　東京都新宿区歌舞伎町2-17-7 松宝ビル
(３月20日本社移転
新宿区西新宿6-15-12　ストーク中央公園９Ｆ)

パソコンワールド

〒103中央区八重洲
1-6-5 中央ビル4F
TEL(271)7139

〝パソコンインストラクター募集〟

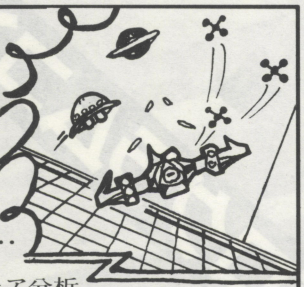
▶募集要項

職種　インストラクター(パソコン)
資格　男子 50歳迄
女子 35歳迄
給与　弊社規定により面接の上決定致します
交通　東京駅八重洲口　徒歩３分
応募　履歴書持参の上ご来社下さい

パソコンレンタルーム

パーソナルコンピュータのあらゆるニーズに対応するパソコンのレンタルームです。ご利用方法はあなた次第。〝気にいったパソコンを自由に操作してみたい〟〝ハード購入前のテストをしてみたい〟〝ソフトを持ち込んでゲーム、ビジネスのソフト作成に使いたい〟〝1ケ月に1度、給与計算等の処理をしたい〟――あなたのアイデアをお待ちしております。

GOTO



研究心
生かしてほしい…

▶事業内容／各種システムソフトウェア分析
・設計・開発　他ソフトウェアパッケージ開発

▶募集要項

職　　　　SE・プログラマー
資　　格　①高卒以上、25歳位迄の男女、コンピュータ技術者になるる意欲あるかた
②来春新卒者(大学・短大・高専・専門)
経験者35歳迄
給　　与　①②未経験者　11万円～15万円
③経験者　18万円～40万円
(経験・能力を考慮の上、優遇)
待　　遇　昇給年１回・賞与年２回・各種社会保険完備
交通費全額支給
勤務時間　９:15～17:15
休日休暇　第１・３・５土曜日・日曜・祝日・年末年始
夏期・特別・有給休暇
応　　募　電話連絡の上、履歴書を持参、又は郵送して下さい
(※入社日、相談応)
交　　通　国鉄お茶ノ水駅下車、徒歩５分

㈱日本ベーシック
〒135　東京都文京区本郷2-9-7 三好ビル
☎ 03(813)4550 担当／福田

株式会社 システムエコー

東京都渋谷区桜ヶ丘15-8 高木産業会館421
〒150　Tel(03)463-3740　担当／武蔵・川上

▶事業内容／カフェテリアシステム・IDカードシステム等のユニークなシステム・ソフトウェアの設計・開発

▶募集要項

未経験者大募集
完全週休２日制

職　　種　SE・プログラマー
資　　格　高卒以上、25歳位迄の
男子未経験者　経験者も可
給　　与　11万円～13万円迄
経験者は高給優遇(例16万円以上)
勤務時間　９:00～17:30
待　　遇　昇給／年１回・賞与／年２回
交通費全額支給・残業手当有
休日休暇　完全週休２日制・年末年始・夏期
有給休暇有
応　　募　電話連絡の上、履歴書(写真貼付)を持参
ご来社下さい。入社日相談に応じます。
交　　通　国電渋谷駅南口、下車５分
㊞銀座線・半蔵門線渋谷駅

君も今日からプログラマー

SKY・HIGH

未経験者歓迎

飛び上がれ CHALLENGER

ゲームだけでは満足出来なくなった君、本格的なビジネス・ソフトを作ってみようと思っている君、もちろん未経験の君でもOKです、一から社会に通用するプロのソフトウェア・マンに育て上げます。

当社では柔軟な頭脳を持ちチャレンジ精神旺盛な君達〝パソコンファン〟の参加を待っています。

当社は超大型コンピューターからパソコンまであらゆる顧客ニーズに対応したソフト開発を行っています。特に外国為替システムでは業界No.1と言われており、地震波の測定システム、ネットワークシステムにも当社の力が発揮されています。昨年はパソコンの事務処理開発支援システム〝STAGE〟を開発し、パッケージソフトの研究・開発・販売にもフィールドを拡げています。これから増々発展していく当社の次代を担うフレッシュな君達を必要としています。

我々と一諸にソフトウェア時代の明日を作ってみようと考える方、気軽に当社を訪ねて下さい。

▶募集要項

職種	プログラマー
資格	高卒以上、未経験者25歳位迄
給与	11万円〜15万円
待遇	昇給年1回・賞与年2回　交費全給 社保完備　海外研修制度　各種クラブ活動
勤務時間	9:00〜17:30（第2・4土曜は9:00〜15:00）
休日	日・祝日　第1・3・5土曜日
休暇	夏期、年末年始、有給休暇
応募	電話連絡の上、履歴書(写真貼付)郵送 入社日相談応ず
交通	国電恵比寿駅下車徒歩1分

株式会社
コンピュータ・コミュニケーションズ
東京都渋谷区恵比寿南1丁目4番2号紀国屋ビル3F　〒150
電話(03)793-0281(代表)

中央システム株式会社

〒160 東京都新宿区大久保2−1−9 第8松田ビル11階
TEL (03)202−4511

一緒にやろうよ

新宿

君もプロになれる!!

未経験者 大募集

職種	プログラマー(将来はSEに!!)
資格	高卒・専門・短大・大卒　25歳位迄
給与	12〜14万円上　（親切丁寧に指導します）
待遇	昇給／年1回 賞与／年2回 社保完 交通費全給
勤務時間	9:00〜17:30
休日休暇	週休2日制 有給休暇 夏季・年末年始休暇・特別各休暇有
応募	電話連絡の上、履歴書持参ご来社下さい。 (履郵送可)　※入社日相談に応じます。
交通	新宿・新大久保・早稲田から徒歩5〜10分

システムライン

中央区日本橋小伝馬町
7-12セントラルビル4F
TEL(639)9656 〒103

大型からパソコン用のソフトまでのシステム・プログラムの設計・開発・分析に基づく経営コンサルタントを行っています。特にBASIC言語を使用した「金銭出納帳」「実用統計分析パッケージ」などは発売以来大好評、さらにソフトのバージョンアップを図るために、パソコンマニアを募集します。

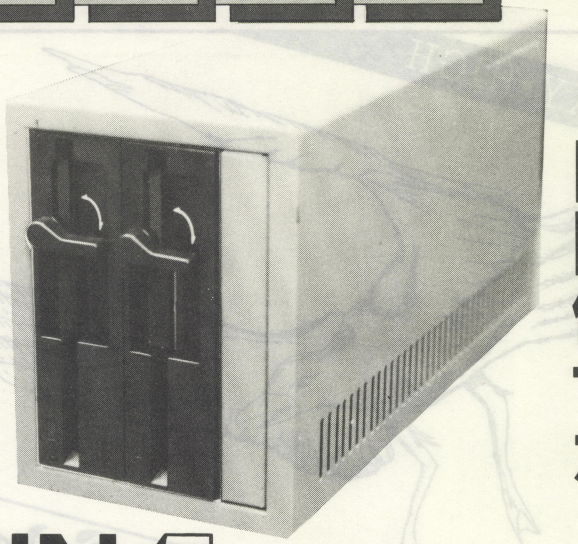
職種	プログラマー(経験なくても可)
資格	高卒以上 25歳迄の男女
給与	11万円〜15万円
待遇	賞与年2回、昇給年1回、各種社保完備 交通費全額支給
時間	9:00〜17:00
休日	隔週土曜、日曜、祝日、年末年始、夏期 特別、有給休暇有り
応募	電話連絡の上履歴書持参ご来社下さい
交通	日比谷線小伝馬町下車徒歩2分

84-4 oh!FM

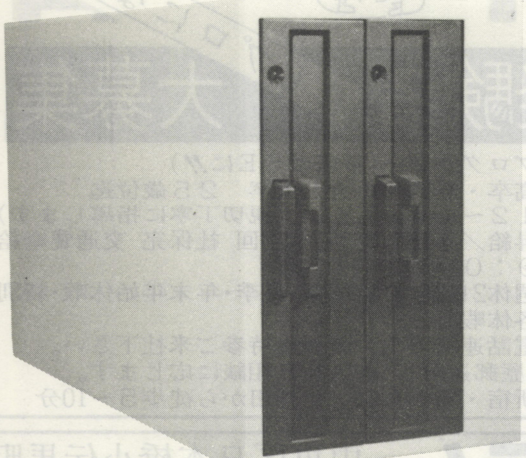
オリジナルドライブ

THIN LINE 8 FM-11
¥268,000

- 富士通純正品と同じYE-DATA社高信頼8インチフロッピードライブYD180-1600を使用
- 電源には8インチディスクドライブ専用に高信頼スイッチング電源を特別設計
- インターフェース別売
- ケーブル付属



FM関係サポート充実！



THIN LINE 5 FM-7
¥135,000

- 富士通純正品と同じYE-DATA社高信頼5インチ薄型フロッピードライブYD-580を使用
- 電源には5インチディスクドライブ専用に特別設計された高信頼スイッチング電源を採用
- インターフェース及びケーブル付属

ADO マイコンショップ 1F ブックショップ 2F
03-255-9515
〒101 東京都千代田区外神田2-2-1

4/1〜4/15 新店舗オープン 1周年記念 セール実施

その1 PC-9801E＋8インチインターフェース＋シンライン8/PCデュアルフロッピー
ナント¥415,000

その2 PC-9801E＋8インチインターフェース＋シンライン8/PCデュアルフロッピー＋シャープ14M-132C（4000文字対応）
限定10台 ナント¥500,000

その3 PC-8801mkIIモデル10＋フロッピー2台
ナント¥205,000

その4 PC-8801mkIIモデル10＋フロッピー2台＋シャープ14M-131C（2000文字カラー）
限定10台 ナント¥255,000

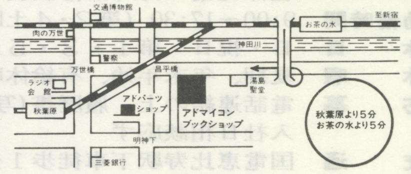
その5 当社オリジナルフロッピーディスクユニット
限定各10台限り 各種15%引

その6 アップルゲームソフト特価奉仕 乞う御期待！

ブックショップ

FM-7 活用研究	¥1,500
FM-7 マシン語入門	¥1,800
パソコンFM-7入門	¥ 680
FM-7/8マシン語ゲームプログラミング	¥2,200
FM-7/8マシン語の本	¥2,800
FMシリーズマイコンが解く天体の謎	¥1,600
FM-7 F-BASIC解析マニュアルフェーズI基礎編	¥2,800
〃 〃 フェーズII探究編	¥2,800
ジ・アドベンチャーFM-7/8	¥2,800
パソコンBASICのグラフィックスFM-7/8	¥2,000
FM-7マイコンアニメーション1	¥2,000
FM-7 FAN BOOKS 1	¥1,200
〃 2	¥1,500
〃 3マシン語入門	¥1,800
FM-7わくわく活用入門	¥2,100
FM-7ユーザーズマニュアル	¥2,500
FM-11 システム解説	¥3,200

シャープ MZ-2000 ¥85,000
×1（G-RAM付） ¥175,000



通信販売ご利用のお客様は下記へお問合せ！もしくは現金書留にてお送り下さい。
〒101 東京都千代田区外神田2-4-6
ビルディングササゲ7F
株式会社 亜土電子工業
通販部 oh!FM係 TEL 03-253-8307
※お願い：住所・氏名・TEL・注文書は明確に、また、お忘れない様！

OAソフト-03 顧客管理プログラム

OAソフト-03は、日本語処理により1件のデータの社名、住所等の固定項目が5項目。その他、任意の項目が最大7項目まで入力できデータ件数は、1000～3000迄可能です。

検索条件により顧客管理表、住所録、電話帳、宛名印刷等が自在に出来ます。

FM-11、AD、EX、AD2、BS専用……………￥38,000

OAソフト-01プログラムレス
言語による各種帳表作成……………￥76,000

ダイナミック株式会社

TEL 0476(24)2423代表　千葉県成田市東和田355番地

FM-7マイコンアニメーション1
《パソコングラフィックス入門》

安居院猛・中嶋正之共著
★B5判・定価2,000円

新しい映像を試みようとする人のために、コンピュータアニメーションの作画技法について、わかりやすく解説。マイコン紙芝居、マイコン絵はがきなど楽しいものから服装と室内配色、モザイクとステンドグラス、ビジネスグラフィックスへの応用など、豊富に掲載。

FM-7マイコンアニメーション2
《3次元グラフィックスの実際》

安居院猛・中嶋正之共著
★B5判・定価2,000円

コンピュータグラフィックスやアニメーションに興味をもつ人のために、使って楽しめるプログラムを中心に解説。キーフレーム法をはじめ立体アニメーションの作り方、立体データの入力方法、透視変換、空間変換、隠面消去や陰影づけなど、各種手法を掲載。

「FM-7マイコンアニメーション1・2」のカセットテープ(3巻1組)および フロッピーディスクも発売しております。《お問い合せ先》(〒150)東京都渋谷区南平台6-17 株式会社JCGLまで

廣済堂産報出版

(〒110)東京都台東区東上野4-25-18
☎03(842)0511(大代)

通産省国家資格。情報処理技術者。

国家試験 対策通信講座

●早い準備で合格へ！今からスタート。あなたも！

★受験予定の方は必読！合格資料を無料呈！

合格資料無料呈！
●ハガキか電話で直接下記までどうぞ！

●情報処理技術者は不足！
コンピュータの技術進歩と設置台数の急増により、情報処理技術者は不足！

●国家試験受験準備講座を開講中！
実戦即応の教材と実戦模試・ポイント指導で合格力を養成。早い準備で合格へ。

●将来性のある情報処理技術者！
企業・官庁で有資格者優遇。学生に最適。

●受験資格は不問。誰でも可！
受験資格については、学歴・年齢・性別・経験などは問いません。試験地は全国9都府市で行われます。4科目の筆記試験。

●第2種情報処理技術者国家試験の合格のポイントは、早い準備と限られた時間をムダなく効率的に活用することです。

●資料希望の方は、ハガキか電話で！

㈶実務教育研究所 受験講座情報係 ☎東京03(357)8153
160 東京都新宿区大京町4の380

新学期に向けて前進！

EXPERT MATHEMATICS

中学数学シリーズ新発売中！

発売中のソフト 各3,000円

学年	タイトル	内容	PC-6001シリーズ	PC-8001シリーズ	FM-7 FM-8	X1・X1D X1C	MZ-700 MZ-1200	MSX
中1	整数 1	約数・倍数・素因数分解	○	○	○	○	○	○
中1	整数 2	最大公約数・最小公倍数・文章問題	○	○	○	○	○	○
中1	正の数・負の数	数直線・加減法・乗除法・混合問題	○	○	○	○	○	
中1	文字と式	文字式・式の値・式の計算	○	○	○	○	○	○
中1	一次方程式1	等式の性質・一次方程式の解法	○	○	○	○	○	○
中1	一次方程式2	文章問題の解法	○	○	○	○	○	
中1	関数 1	関数式とグラフ	○		○	○		
中1	関数 2	比例・反比例	○		○	○		
中1	平面図形 1	基本図形	○		○			
中1	平面図形 2	円・おうぎ形の面積 他	3/20		3/30			
中1	空間図形 1	空間の平面と直線	4/20		4/30			
中1	空間図形 2	体積と表面積	4/20		4/30			
中2	式の計算	式の加減・乗除・応用問題	○	○	○	○	○	○
中2	不等式 1	不等式の解法	○	○	○	○	○	○
中2	不等式 2	連立不等式の解法	○	○	○	○	○	
中2	連立方程式1	加減法・代入法・( )の問題・分数の問題	○	○	○	○	○	
中2	連立方程式2	文章問題の解法	○	○	○	○	○	
中2	一次関数 1	一次関数とグラフ	○		○			
中2	一次関数 2	一次方程式と一次関数	○		○			
中2	図形 1	角と平行線	○		○			○
中2	図形 2	三角形と合同	○		○			
中2	図形 3	平行四辺形と応用	4/20		4/30			
中2	図形 4	相似と応用	4/20		4/30			
中3	式の計算 1	展開と乗法公式	○	○	○	○	○	○
中3	式の計算 2	因数分解とその応用	○	○	○	○	○	○
中3	平方根	√ の計算と応用	○	○	○	○	○	
中3	二次方程式1	二次方程式の解法と応用	○	○	○			
中3	二次方程式2	二次方程式の文章問題	○	○	○			
中3	二次関数	グラフと二次方程式	○		○			
中3	円 1	基本的性質と円周角	○		○			
中3	円 2	内接四角形と接弦定理	3/15		3/20			
中3	三平方の定理1	三平方の定理と平面図形	3/30		5/10			
中3	三平方の定理2	空間図形への応用	3/30		5/10			

ボクの名前募集中！

(官製ハガキで応募下さい。)

採用者の中から抽選にて

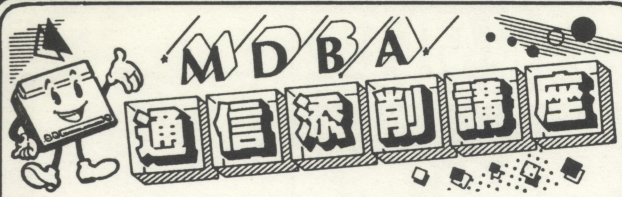
1名に 中学英作文シリーズ 全9巻 (52,200円)
3名に 社会科シリーズ 全4巻 (12,000円)
20名に オリジナルカレンダー

(〆切 4月18日)

代理店募集中！
学習ソフトオールカタログが必要な方は、切手150円分と右のカタログ請求券同封で御請求下さい。

株式会社 数研塾 開発事業部
〒411 静岡県三島市一番町15の32 芹沢ビル4F
TEL(0559)75-1534㈹
振込先：駿河銀行三島駅前支店㊟606912（送料不要）

数研塾 カタログ請求券 Oh!FM 4月号



● 自由な時間にプログラミング，いかがですか？

パソコンを買ったのに，プログラムを習うチャンスがなく，市販ソフトだけでがまんしている方，自習書などで勉強しているけれども，よくわからない方，歓迎いたします。

● 実技中心の講座です

プログラミング技術は，何本も何本もプログラムを作ることにより上達する技術です。当講座は，出来るだけプログラミングをして頂くよう，ワークカセットを用意し，当社に送られてきたプログラムは，プリンタからリストを出力し，添削して返送する出題形式等，工夫をこらしたカリキュラムになっています。

● 幅広い学習が期待できます

当社はゲームソフトから簡易言語，データベース，コンピュータ・グラフィックス・ソフトまで開発してきたシステムハウスですから，幅広いプログラミング技術が学べます。

● プログラム作りには良い道具が必要です

ご入会の方には，プロが使っている，使い易く，保管し易いMDBA特製仕様書類及びバインダー，テンプレートを使って頂きます。良いプログラムを作るには，良い道具が必要です。あなたの部屋も明日からはシステム開発室です。

対象機種 ①PC-8001mkII,PC-8001
②FM-7
③X-1
④PC-6001

（MSX用も開講準備中）
※機種別の講座ですから，あなたのパソコンにあわせた学習が行なえます。

講座 ①ビギナー講座
初めての方、基礎から始めたい方を対象。

②初級講座
FORループが使える方を対象

バグカード　システム

自作プログラムでどうしてもわからないバグにお答えいたします。

※実技中心の講座のため，ご入会はパソコンをお持ちの方に限らせて頂きます。

資料希望の方はハガキに〒,住所，氏名，TEL，機種名を記入の上，当社宛にお送り下さい。

マイクロ データ ベースアソシエイツ

〒194-01 東京都町田市鶴川2-1-8 -205

☎(0427)35-1600

FM-7 NEW DOS TURBO

商品番号：F10，機種：FM-7，価格：15,000円

FM-7に画期的DISK BASIC誕生！

★プログラムのセーブ時間を半分に短縮
★富士通DISK BASICと同機能（すべてのディスクコードを使用可能）
★プログラムファイル、シーケンシャル、ランダムファイルなどすべての機能をサポート
★ディスクドライブ4台、ファイルのオープン数15まで設定なしに使用可
★立ち上がり時のオートラン、ファンクションキー定義可
★NEW DOS TURBO, NEW DOS、DOSモードDISK BASIC、富士通DISK BASICのディスクはコンパチブル
★プロテクトなし（IPL、ディスクコードの研究にも利用ください）
★\$7000～\$7FFFはフリーエリアとなり、30530バイトを自由に使える
3月31日まで4本の実用プログラムサービス！
1.DOSモードDISK BASIC
2.スクリーンエディット式機械語モニタ
3.機械語テープアトリビュートサーチ　4.ディスクダンプ

FMシリーズ

商品番号	商品名	内容	機種	価格
F1	NEW DOS	フリーエリアはROM BASICと同じ。プログラムファイルのみサポート。\$7000～\$7FFFのセーブ/ロード可。	FM-7、8	F1+F2セット ¥10,000
F2	DOSモード DISK BASIC	DOSモードで使用可能なDISK BASICを作成する。ディスケットは、富士通DISK BASICとコンパチ。		
F3	6809 アセンブラ	本格的2パスアセンブラ。ディスクまたはメモリ上のソースをアセンブル。表記はモトローラ方式	FM-7、8、11	¥ 8,000
F4	6809 逆アセンブラ	1パス、2パス逆アセンブラ。オフセット、DPレジ、データエリアの設定可。ラベル付で逆アセンブル結果を出力。		¥ 8,000
F7	ファイル変換	ディスケットをF・BASICとCP/M-80、CP/M-86で使用できるようにFATとディレクトリを変換する。		¥ 6,000
F8	6809 デバッガ	会話型デバッガ(CP/M DDT準拠)。1パスアセンブラ、逆アセンブラ、BPセット、リセット、メモリの16進/アスキダンプ、Fill、Move、オフセット計算レジスタ表示/修正、メモリライト、ユーザープログ実行、トレース(1～FFFFHステップ)		¥12,000
F9	ファイルコピー	NEC PCシリーズでアスキーセーブされたディスケットを、FM用にコピー。PC1W、2Wともに可。		¥ 6,000
F10	NEW DOS TURBO	新製品、上記参照、新発売記念セール中	FM-7	¥15,000

CP/Mシリーズ ＊CP/Mはディジタルリサーチ社の登録商標です

商品番号	商品名	内容	機種	価格
C1	Z80アセンブラ (ZAS.COM)	ザイログ形式Z80ニモニックをアセンブルし、HEXファイルを作成。使用方法はASM.COMと同じ	下記①	C1+C2セット ¥12,000
C2	Z80逆アセンブラ (ZDA.COM)	2パス方式Z80逆アセンブラ。ラベル付出力、ソース形式、プリント形式ファイルを作成。オフセット設定により正常な逆アセンブルを行なう		
C3	ファイル転送 (TFR.COM)	COMファイル、1AHを含むファイルを初めとするすべてのファイル形式を、2つのCP/Mシステム間で転送		¥ 6,000
C5	フォーマット変換 (YOU.COM)	各社パソコンのCP/M用ディスケットをドライブB:でリード/ライト可能にする。 ディスケット： CP/M-80用 FM-7/8/11、PC1W/2W、パソピア、MBC100、IF800、MZ80B、マイクロデジジョン、FP1100、QC10 CP/M-86用 FM-8/11、PC9801、マルチ16	下記②	¥16,000
C6	6809デバッガ (DEBUG.COM)	F8(6809デバッガ)のCP/M-80版、FMシリーズならではの6809版CP/Mとでも云うべき機能発揮	FM-7、8、11	¥12,000
C7	ラムディスク-80 (RDXX.COM)	FM-11メインRAM64KとV・RAM192Kをディスク(ドライブD)として使用可能にする。SUBMIT可。CP/M-80用	FM-11	¥16,000
C9	ラムディスク-86 (RD86.CMD)	FM-11拡張RAMをディスク(ドライブC)として使用可能にする。128K～768K。CP/M-86用		¥16,000

機種名：①FM-7/8/11，PC2W，パソピア，FP1100，IF800
②①の他にQC10

★ご注文は、現金書留または銀行振込にて販売元へお願いします。（送料無料）
★ご注文の際は、商品番号、商品名、使用する機種名、住所、氏名、TELをお知らせください。
★メディア：5.25″ディスケット

販売元　**カーネル**
〒130 東京都墨田区横川3-1-5 石野ビル201
TEL 03—624—4921
振込口座　第一勧業銀行　押上支店(普)1189195
開発　**橋口技研**

EDITOR'S ROOM

◆FMのハードやソフトに詳しい人で，原稿の書ける方を募集しています。電話のうえ，実力評価用の作品をご持参ください。また地方の方は，簡単な自己紹介，マイコン歴，得意な分野などの資料とともに，作品をご郵送ください。グループ応募も歓迎いたします。

〒102 千代田区四番町2-1 ㈱日本ソフトバンク出版部
Oh/FM編集室 松岡 ☎03(261)4095

◆本誌の内容に関して，電話による質問を受け付けます。

☎03(265)5789 受付時間 月〜金 16：00〜18：00

質問はバグ情報についてのみ受け付けます。機械の取り扱いなどについてのお問い合わせはご遠慮ください。

▶広告索引 五十音順

アーバン電子……143
アイ企画……155
アイテム……154
アコリット……150
亜土電子工業……164
エイト電気……148
エス・ピー・シー……144・145
エプソン……表紙3
エム・アイ・エー……151
エム・エー・シー……カラー16
岡田商事……カラー17
カーネル……167
関東電子……カラー6・7
クリスタルソフト……カラー13
光栄マイコンシステム……146・147
廣済堂……165
コマス……カラー14
コンピュータ・イレブン……157
コンピューター応用開発……143
シー・シー・ダブル……153
J&P……カラー10・11
実務教育研究所……142・166
新紀元社……149
数研塾……166
ソフトマート……158
ダイナミック……165
チャンピオンソフト……カラー12
九十九電機……160
テクノソフト……カラー15
日本情報システム……90
ハドソン……カラー8・9
ピー・シー・エー……159
富士通……表紙2・4，カラー1・2・3
プログラマー募集……162・163
ボーステック……161
ポリシー……156
マイクロデータベースアソシエイツ……167
明光製作所……152
理工産業……142

▶編集後記

■月刊第1号の表紙は，㊍自ら絵筆，いやキーボードをふるいました。しかしみなさん，表紙を見てその作者の生活を想像してはいけませんよ。今，ボクは試験中なんです。こんな生活はもういやじゃ。早く青い空の下，ドライブに行きたい。しかし，そうもいかないんですよね。試験が終わったら，再び，マリちゃん(M)の目が光り出しますから。 ㊍

■なん度も降り，あちこちに消えずに残る雪の中，期末試験で久しぶり(?)に勉強に励むころ書く編集後記。これが活字になるころは雪も完全に消え，僕は春休みの旅行を楽しみ，受験生だった人も明るい生活に戻ることと思います。もしこの春入学祝でなにか買ってもらえそうな人がいたらFM-7を買わせ，Oh/FMを愛読させましょう。すると，月刊化も果たしたOh/FMは驚異的に売り上げを伸ばし……。 (ゑ)

■今年の東京はどうなってるんでしょう？ 大雪で道路は凍るし，とっても寒いし，このシーズンに試験責めにあっているこちらの身をもっと考えてほしい……。今月は，お茶くみ専門でした。 Ⓨ

■テストが終わったと思ったら，原稿の嵐…。特集はいかがでしたか？ もっともっと細部まで追究したかったのですが，持病のアル中が再発して書けんかった（ゴメンナサイ/） ⑪

■やったね/ついに復帰しました。この記念にと編集部に後輩数人を連れてきました。この中の1人♨君先日のバレンタインに同じ身上の¥君と下宿で酒を飲むことにしたんだそうです。そこで近所の店に酒を買付けにいったところ店のおばさんがチロルチョコをくださったのだそうです。二人はとても喜んだそうです。☺

■来月から本格的に参加するつもりで，ちょこっと顔を出したのが運のツキで，蛸部屋Oh/FMにどっぷりと首までつかってしまったKazです。原稿書きには慣れていませんが，頑張るつもりですのでヨロシクお願いいたしま〜す。 (Kz)

■「ブレインストーム」試写会報告。監督はEEGのリーダーで，「2001年宇宙の旅」「ブレードランナー」などの特撮を担当したダグラス・トランブル。人間の思考・意識などを再生，他人も同体験できるというマシンをめぐっての話だけど，みものは特撮。観客もそのマシンをつけているような気にさせられるくらいの臨場感。正直「ブレードランナー」よりボルテージは落ちている感じ。でも単なる絵空事とは思えないリアル感に，思わず拍手。 (妙)

■突然ですが，ボクは“ラー中”といいまして，1日1回は必ずラーメンを食べないと気がすまないラーメン中毒者なのデス。寒い時のラーメン…最高だね。おいしいラーメン屋さんがありましたら，ぜひ教えてください。あっ，自己紹介が遅れました。月刊になってのニューフェースです。ヨロシク!! (谷)

■月刊第1号はいかがでしたか。今まで隔月刊だったため連載ものはできるだけ避けていましたが，そのうっぷんを晴らすかのように連載ものが一杯という結果になってしまいました。

さて，5月号はワープロ特集，6月号は音楽特集となります。YMOの曲など自慢の作品をお寄せください。短くてもガッツの入った作品をお待ちしています。 (M)

・編集 松岡真理 池谷芳信 真部一善 渡辺妙子
・技術 松田辰夫 小林初雄 徳永 聡
・海外協力 HONG・LIANG・LU IAN・ALLEN
RONALD・BILLINGS HISAYOSHI・MIKAMI
・協力スタッフ 川西慶一 菊池蓉子 木下淳博 志沢雅彦
末安直樹 鶴岡哲明 長沼孝仁 西村義孝 平野 剛
古田 昌 本郷誠司 米村英明 林 剛正
・写真 浜崎 昭 杉山和美
・レイアウト オメガ社
・アートディレクター 藤林省三
・表紙 Tea Spoon 木下淳博

月刊 Oh!FM 4月号

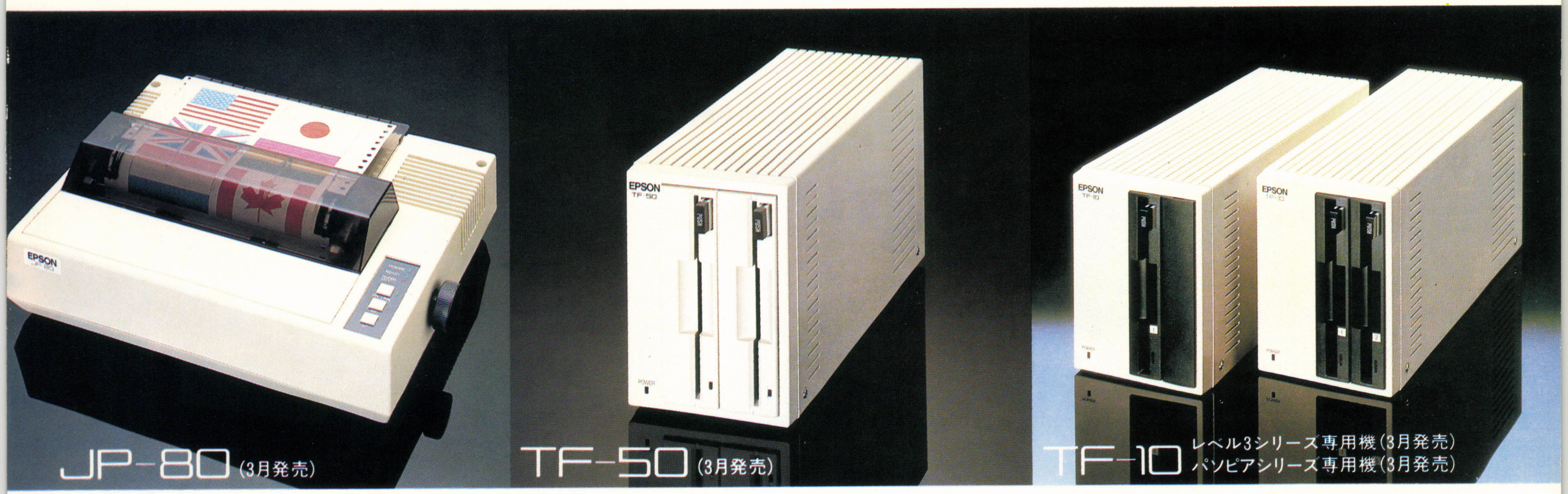
・昭和59年4月1日発行
・発行人／孫 正義
・編集人／田鎖洋治郎
・発行元／株式会社日本ソフトバンク
本社／〒102 東京都千代田区九段南2-3-14 靖国九段南ビル2F
☎03-263-3690 Fax. 03-263-3660
出版部／〒102 東京都千代田区四番町2-1 ☎03-261-4095
大阪支店／〒542 大阪市南区難波千日前5-19河原センタービル
☎06-644-0191 Fax. 06-644-0160
・印刷／図書印刷 ☎03-453-2550
・海外送付ご希望の方は下記へお問い合わせください。
日本IPS株式会社／〒101 東京都千代田区神田小川町3-5 ☎03-291-2632(直)

©1984 JAPAN SOFT BANK 雑誌02199-4
★本誌からの無断転載を禁じます。

EPSON

ペリフェラルはエプソン

ニーズを極めて新登場。充実のラインナップ。



多彩な機能を搭載、7色のカラー指定が可能なドット・マトリクス・プリンタJP-80新登場。●印字速度160cps、双方向最短距離印字(ロジカルシーキング)テキスト時、単方向印字(左→右)ビットイメージ時●4″~10″のファンフォールド紙、レター用紙が使える紙幅可変スプロケットフィード、フリクションフィード●アンダーライン機能、スーパー/サブスクリプト文字、国際文字印字(11ヶ国対応)、ダウンロード文字(ユーザー定義可能文字)、さらにプロポーショナル印字も可能。
■印字速度:160字/秒(普通文字)■用紙:ファンフォールド紙(4″~10″)■コピー:1オリジナル+2コピー(但し紙厚0.3mm以下)■改行時間:約150ms(1/6″=1行時)■印字方向:通常文字/双方向最短距離印字(ロシカルシーキング)、スーパー/サブスクリプト・ビットイメージ/左→右の単方向印字■文字種類:JIS128文字 +ASCII 小文字31文字+グラフィック63文字+国際文字32文字(11ヶ国対応)+ダウンロード文字(ユーザー定義可能文字)256文字■印字方式:インパクト・ドット・マトリクス(9ピン)■超高精度ビットイメージモード9種類■外形寸法:441W×347D×115Hmm(トラクタユニット装着時155Hmm)■重量:約7.5kg

パソコンを手軽にパワフルにシステムアップ。大容量ターミナルフロッピーTF-50新登場。●5.25インチでありながら、標準フロッピー(8インチFDD)と同一の大記憶容量3.2Mバイト(1台当り・アンフォーマット時)を実現。●両面高密度ミニフロッピーディスクの採用により、高信頼性・高性能・大記憶容量を確保し、しかも大幅にコストダウン。●エプソン日本語ワープロパソコンQC-10、10IIおよびPC-9801/8801/8001mkII等のパソコンとの接続により、標準フロッピー(8インチFDD)の増設ドライブとして、また5インチFDDと併用した大容量データディスクとして使用が可能。●TFシリーズのデザインを受け継いだ場所をとらないコンパクトな薄型設計。
■記憶容量:3.2Mバイト(1台当り・アンフォーマット時)2.05Mバイト(1台当りフォーマット時)■1ドライブ当りのトラック数:154トラック■1トラック当りのセクタ数:26セクタ■1セクト当りの記憶容量:256バイト■電源電圧:AC100V±10%■電源周波数:49.5~60.5Hz■消費電力:40VA以下■外形寸法:120W×350D×165Hmm■重量:7.5kg

幅広いユーザーニーズに応える、高性能・低価格ターミナルフロッピーTF-10。新たなパソピアシリーズ専用機、レベル3シリーズ専用機を加え充実のラインナップ。●PCシリーズ、FMシリーズ、MZ・FPシリーズ、レベル3シリーズ、パソピアシリーズ等の各種専用機をラインナップ。●オプション基板を取り換えるだけで、各種のパソコンに対応可能。●各機種ともデュアルタイプ・シングルタイプ構成。
■記憶容量(両面倍密度記憶方式の場合):640Kバイト(デュアルタイプ)320Kバイト(シングルタイプ)■1ドライブ当りのトラック数:80トラック(40×2)■1トラック当りのセクタ数:16セクタ■1セクタ当りの記憶容量:256バイト■電源電圧:AC100V±10%■電源周波数:49.5~60.5Hz■消費電力:40VA以下■外形寸法:120W×350D×165Hmm■重量:7.0kg(デュアルタイプ)5.85kg(シングルタイプ)■価格:PCシリーズ専用機¥129,800(デュアルタイプ)¥89,000(シングルタイプ)、FMシリーズ専用機¥125,800(デュアルタイプ)¥85,000(シングルタイプ)、MZ・FPシリーズ専用機¥122,800(デュアルタイプ)¥82,000(シングルタイプ) 〈オプション〉PCシリーズ専用基板¥29,000、FMシリーズ専用基板¥25,000、MZ・FPシリーズ専用基板¥22,000、増設ドライブ¥45,000

あなたのパソコンとリンクする、ペリフェラルはエプソン。

ターミナルプリンタ

〈RPシリーズ〉

RP-80 ●MPの技術が生んだ高性能の経済派プリンタ¥89,000

RP-80F/T ●MP-80F/TIIIのシステムアップとコストダウンを実現した高精度ビットイメージプリンタ¥104,000

RP-100 ●RP-80の機能に加え、インターフェイスを充実し、汎用性を高めたビットイメージプリンタ¥134,000

〈FPシリーズ〉

FP-80 ●高速印字と多彩な文字バリエーションの強力インテリジェントターミナル¥149,800(PC-8001専用¥152,800/PC-8001mkII、8801、9801専用¥153,800)FP-80用漢字ユニット#8175¥40,000(JISコードで漢字プリンタとして動作。)

FP-80K ●多彩な機能を搭載した高品質漢字プリンタ¥189,800(PC-8001専用¥192,800/PC-8001mkII、8801専用¥193,800)

FP-100 ●トータルバランスに秀れた万能型スーパープリンタ¥189,800(PC-8001専用¥192,800/PC-8001mkII、8801、9801専用¥193,800)

UP-130K ●印字速度を飛躍的にアップ、多機能漢字プリンタUP-130K。¥290,000(JIS漢字第1水準)¥310,000(JIS漢字第2水準)

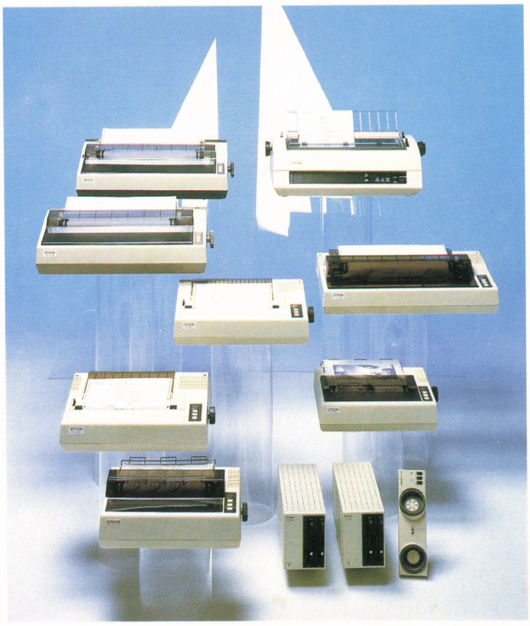
DP-100 ●多彩な機能を搭載したデイジーホイールプリンタ¥140,000(パラレル仕様) ¥150,000(シリアル仕様)
〈オプション〉トラクタフィーダー¥24,000、オートカットシートフィーダー¥50,000、キーボードユニット¥39,000

ターミナルフロッピー

TF-20 ●APPLE II専用機デュアルタイプ¥128,000・シングルタイプ¥83,000(シングルタイプはオプションの増設ドライブを加えることでデュアルタイプとなります。¥50,000)

パーソナルカプラ

CP-20 ●コンピュータネットワーク、データ通信を強力にサポート。¥49,800



好評発売中	プリンタ操縦法入門	EPSON FP-80(RP-80・MP-80)活用マニュアル ¥2,400/AB版224ページ/ラジオ技術社発行/東京電機大学助教授 脇 英世編著

エプソン株式会社 ●本社/〒399-07長野県塩尻市広丘原新田80☎0263-52-2552(代) **エプソン販売株式会社** ●本社/〒160東京都新宿区西新宿2-4-1新宿NSビル内私書箱6109号☎03-348-7121(大代)・大阪支店/〒530大阪市北区西天満4-14-3住友生命御堂筋ビル11階☎06-365-5071・札幌営業所/〒060札幌市中央区北一条西2丁目札幌時計台ビル6階☎011-222-2821・秋田営業所/〒010秋田市中通2丁目176秋田東京火災海上ビル7階☎0188-32-4002・仙台営業所/〒980仙台市本町1-12-12山万ビル6階☎0222-63-3691・長野営業所/〒390松本市深志2-2-9加納ビル2階☎0263-36-7251・新潟営業所/〒950新潟市東大通1-2-25北越第1ビル6階☎0252-43-8515・名古屋営業所/〒460名古屋市中区新栄町2-13栄第一生命ビル9階☎052-962-7001・金沢営業所/〒920金沢市彦三町2-3-13金沢彦三ビル4階☎0762-62-3216・広島営業所/〒730広島市中区幟町3-1第3山県ビル4階☎082-223-2811・福岡営業所/〒812福岡市博多区博多駅前3-4-8郵政互助会博多ビル4階☎092-471-0761・鹿児島営業所/〒892鹿児島市加治屋町18-8三井生命鹿児島ビル4階☎0992-25-7717●詳しい資料のご請求は、はがきに住所、氏名、年令、職業、製品名をお書きの上、もよりのエプソン販売までお申込みください。

ペリフェラル
資料請求券
OH!FM

Oh!/FM 1984 4月号
昭和五十九年四月一日発行 第二巻第二号通巻八号
昭和五十九年二月二十八日国鉄首都特別扱承認雑誌第七四〇三号

信頼と創造の富士通

富士通

FM SPIRIT.
実感、超えてる人のビジネスごころ

富士通のパソコンFM-11、バージョンアップして新登場。

最新のOAを、まるでペンを走らせるように気楽に使いこなしてしまう
そんなビジネスマンが増えています。
先端技術を有能なパートナーに、
そんな粋なFMスピリッツが、今、注目を集めています。

FM-11 イレブン
新 登 場

1MBの大容量5インチFDを2基搭載
BS ¥398,000 (本体価格・日本語ワープロソフト付)

OS-9を採用、ビジネスもマルチ感覚
AD2 ¥298,000 (本体価格・日本語ワープロソフト付)

●写真はFM-11 BSです。(CRTは新タイプMB27315)

富士通株式会社 半導体統轄営業部 〒105 東京都港区虎ノ門2-3-13 ☎(03)502-0161

Oh!/FM ㈱日本ソフトバンク発行 Printed in Japan 定価480円 雑誌 02199-4

Oh!FM 1984 4

●特集=虫取り専科

●FM-7を解析する

●機械語マスタ'84

日本ソフトバンク